UTILITY OF EXPERIMENTAL DESIGN IN AUTOMATIC TARGET RECOGNITION PERFORMANCE EVALUATION

THESIS

James M. Higdon, Captain, USAF

AFIT/GOR/ENS/01M-08

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/GOR/ENS/01M-08

UTILITY OF EXPERIMENTAL DESIGN IN AUTOMATIC TARGET RECOGNITION PERFORMANCE EVALUATION

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

James M. Higdon, B.S.

Captain, USAF

March 2001

AFIT/GOR/ENS/01M-08

UTILITY OF EXPERIMENTAL DESIGN IN AUTOMATIC TARGET RECOGNITION PERFORMANCE EVALUATION

James M. Higdon, B.S.
Captain, USAF

Approved:

Kenneth W. Bauer (Advisor) — 22 FEB 01 (date)

Jeffrey W. Lanning (Co-advisor) — 22 Feb 01 (date)

Acknowledgements

It has been a privilege to work with my advisor, Dr. Kenneth Bauer, and my co-advisor, Maj. Jeffrey Lanning. I thank them for their guidance, understanding and their generous opinion of my aptitude. Thanks also to my sponsor, Dr. Timothy Ross for the support and the latitude I enjoyed during my research. I thank the faculty for their dedication and genuine concern for my education, and I also thank the staff for all their helpfulness.

I thank my parents who influenced my decision to pursue the study of mathematics. I thank my children who provided me with the motivation to excel and the perspective to enjoy this experience. I am deeply indebted to my gracious and patient wife who struggled through this program as often as I did, and can legitimately claim to have earned this degree with me. Thanks also for all the morale, recreation and welfare support. Finally, I thank God for the opportunity to make myself a better analyst, officer, and person.

Table of Contents

List of Figures

List of Tables

Abstract

This research investigates current practices in test and evaluation of classification algorithms, and recommends improvements. We scrutinize the evaluation of automatic target recognition algorithms and rationalize the potential for improvements in the accepted methodology. We propose improvements through the use of an experimental design approach to testing. We demonstrate the benefits of improvements by simulating algorithm performance data and using both methodologies to generate evaluation results. The simulated data is varied to test the sensitivity of the benefits to a broad set of outcomes.

The opportunities for improvement are threefold. First, the current practice of "one-at-a-time" factor variation (only one factor is varied in each test condition) fails to capture the effect of multiple factors. Next, the coarse characterization of data misses the opportunity to reduce the estimate of noise in test through the observation of uncontrolled factors. Finally, the lack of advanced data reduction and analysis tools renders analysis and reporting tedious and inefficient. This research addresses these shortcomings and recommends specific remedies through factorial testing, detailed data characterization, and logistic regression. We show how these innovations improve the accuracy and efficiency of automatic target recognition performance evaluation.

UTILITY OF EXPERIMENTAL DESIGN IN AUTOMATIC TARGET RECOGNITION PERFORMANCE EVALUATION

1. INTRODUCTION.

The focus of this research is the application of existing statistical techniques to improve the test methodology for a military organization. We review current practices in the field of automatic target recognition (ATR) performance evaluation and present recommendations for improvement. We support our recommendations by explaining each improvement and simulating the impact. Our primary objective is demonstrating the potential for improvement in test results using our recommended methodology.

1.1. Automatic Target Recognition

ATR is the field of using computer programs to automatically recognize objects of military interest. The military relies on electronic sensors to recognize objects on the ground and in the air for the purpose of targeting and mission planning. These sensors collect images of objects of interest using a variety of different media. Some sensors collect electro-optical images, others collect radio frequency (RF) or infrared (IR)

images. Objects within these images are classified by computer algorithms and the classification performance results are evaluated.

1.1.1. Automatic Target Recognition Algorithms.

Algorithms are designed to locate unique features within an image and associate those features with specific military systems (one avenue of classification could be: Vehicle, tank, T-72). The field of ATR algorithm development has been advanced extensively and hundreds of algorithms exist that use a variety of techniques to recognize potential military targets [15; 20]. Different algorithms exist that accept images in the same medium and are intended to perform the same task. It is in the best interest of the military, therefore, to select an algorithm whose classification performance exceeds that of others in the same class.

1.1.2. Automatic Target Recognition Algorithm Performance.

Algorithm performance is measured by an algorithm's success in correctly classifying objects in an image database. The output of an algorithm (in ATR, classification of specific military systems) is analyzed by counting the number of successes it realizes in detecting a target and dividing this number by the number of targets of the same type (in an image database). This yields an estimate of the probability of detection. Performance can also be measured by counting the number of declarations of a target when the target is not present (false alarm) and dividing false alarms by the number of objects that could potentially be confused with the target (confusors). This yields an estimate of the probability of a false alarm. Other measures are the probability of correct identification (given a successful detection), or the false alarm rate per unit of

area (given a confusor density in the area). Using such measures, algorithm performance can be compared to a baseline or some other algorithm for the purpose of evaluation.

1.1.3. Automatic Target Recognition Algorithm Performance Evaluations.

There are organizations devoted to the purpose of evaluating competing classification algorithms. In evaluating ATR algorithms, an algorithm is commonly treated as a black box and the analyst evaluates the ability of an algorithm to accomplish its intended purpose (detection, location, classification, identification) [13] by measuring the program's output. Here, we focus on performance evaluations that compare two algorithms.

1.2. Test and Evaluation

Air Force guidance for test and evaluation [9; 10] prescribes the scientific method to conduct government testing. Within this framework, we identify the three phases of testing that address the test *methodology* (see Figure 1.1). The three phases are below.

- Phase 1: Test Design
- Phase 2: Data Collection
- Phase 3: Data Analysis

The statistical techniques we use in these phases determine the methodology for a test. For ATR, we identify the tasks performed in each of these phases. Techniques that determine how we accomplish these tasks are the *components* of our total methodology.

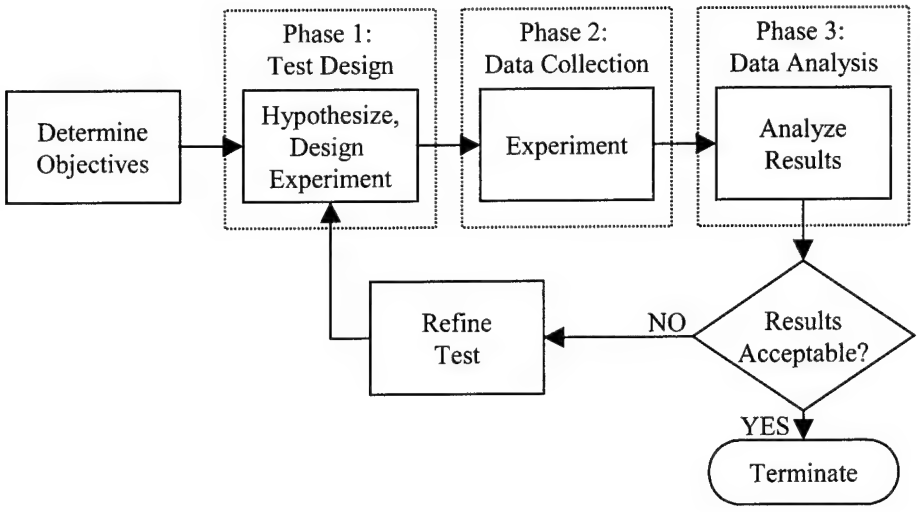

Figure 1.1 The Scientific Method and the Phases of Test and Evaluation

1.2.1. Phase 1: Test Design.

In the design phase of testing, we determine the conditions that we must collect to answer test objectives. Techniques or approaches that address the method for generating test condition matrices are the test design components of our methodology. This phase is complete when we have a set of conditions for which we will collect data for analysis.

1.2.2. Phase 2: Data Collection.

The data collection phase involves the accumulation of data under each of the conditions identified in phase 1. The data collection components of our methodology are the schemes and techniques we use to determine what information is gathered and how it is gathered, given a condition matrix. This phase is complete when we have collected the desired information for each of our test conditions.

1.2.3. Phase 3: Data Analysis.

In the final phase, we use the information we collect in phase 2 to answer our test objectives. Data analysis components of our methodology are the statistical techniques we use to reduce, analyze, and hypothesize about our data. This phase is complete when we have sufficient understanding of test phenomena to answer the objectives.

The combination of all the methodology components from each phase makes up our total test methodology (see Figure 1.2). The numerous techniques available to us in each phase imply that there are many methodologies which can be used to answer the same objectives. In our research, we find that current methods in ATR performance evaluations can be improved.

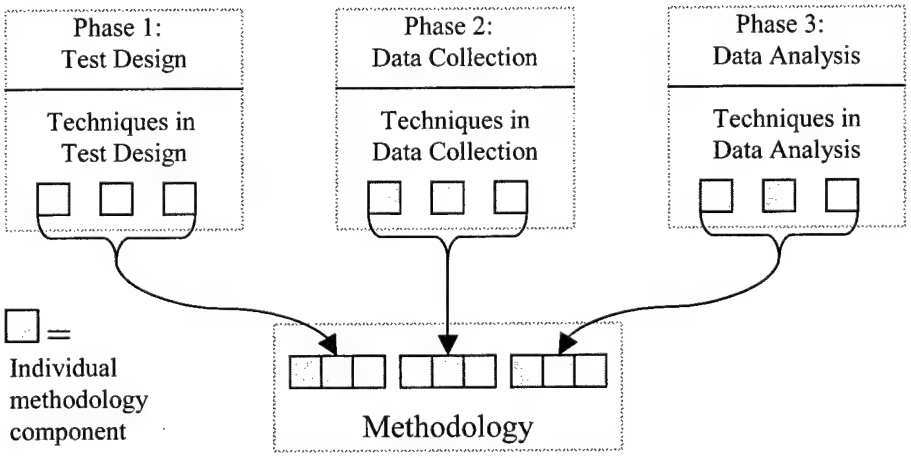

Figure 1.2 Composition of a Test Methodology

1.3. Potential for Improvement in Performance Evaluations

In each of the phases of testing we describe and critique the current evaluation practices in the ATR field. These practices (components) comprise the *current methodology*, and for each component, we recommend a different technique if it yields the potential for improved accuracy, precision, or efficiency. The recommended techniques are the components of our *improved methodology*. We believe the field of ATR performance evaluations is an excellent candidate for standard OR tools and statistical techniques such as factorial test design, iterative data collection, and logistic regression. In ATR performance evaluations, our challenge is to demonstrate the utility of these components of an experimental design approach, or, *paradigm*.

1.4. Research Objective

Our research objective is to show the utility of using experimental design (our improved methodology) in ATR performance evaluations without implementing the improvements in a real test scenario. Instead, we demonstrate the potential for improved results by devising a simplified, yet representative ATR evaluation scenario and estimating the impact of not using experimental design. By applying the concepts of experimental design, we can identify specific improvements for each phase of testing; our recommendations are listed below.

- Test Design Phase: Use factorial design to generate test conditions and use fractionation to manage (reduce) the test condition matrix.
- Data Collection Phase: Employ an iterative collection scheme and increase the detail of image characterization.

- Data Analysis Phase: Use logistic regression to reduce and analyze test data, and utilize hypothesis testing to answer test objectives.

The experimental design paradigm can be used to identify many more potential improvements, but the recommendations above address the most critical deficiencies in the current methodology. The improved methodology, which is based on the experimental design approach to testing, improves our ability to compare algorithms. In addition, there are advantages for other test objectives such as evaluating the possibility of transition to the field (a major undertaking [7; 8]) which requires the broadest set of operating conditions against which to evaluate an algorithm [17; 23]. Now we turn to the compass of our research.

1.5. Research Scope

There are potentially other phases in testing that need development, other potential improvements, and more complex experimental design concepts. In this research, we only investigate the three phases of testing we have identified, we only implement the recommendations listed above, and experimental design is only explained in the detail required to justify our recommendations. Our scope is limited first by these factors. We perform our research in this framework and develop an approach to achieve our test objective.

1.5.1. Ideal Scope of Research.

The ideal approach for demonstrating the utility of experimental design is to collect new data using both experimental design and the current method, then compare the results from each experiment. If we perform this comparison many times, under

many real test scenarios (with different results) we can eventually establish the superior methodology. Unfortunately, there are many obstacles to this approach.

1.5.2. Actual Scope of Research.

Since the opportunity to collect new data is not available, we use theoretical data that is representative of the phenomena encountered in ATR evaluations. Even after comparing the current methodology to our improved methodology, our task is not complete. We still must demonstrate the results of the comparison hold in the face of different datasets. We can accomplish this by comparing the current and improved methodologies against a series of data sets that span the spectrum of possible outcomes for our scenario. This is a formidable task, so we simplify the comparison to make the benefits easily apparent while using an example test scenario that reflects the same issues and objectives faced in real testing.

1.5.3. Outline of Research Approach.

Here we identify our basic approach and introduce a few methodology concepts. The main chapters of our research and the issues covered in each are listed below.

- Chapter 2, review of ATR performance evaluations. In this chapter we briefly describe ATR evaluation methodology and introduce our recommended improvements. We also review research in the fields of ATR and statistics that is relevant to our recommendations.
- Chapter 3, current practices in the evaluation cycle and proposed improvements. Here we describe the components of the current methodology, identify key deficiencies, and explain our recommended improvements for each of the three

phases of testing. We give detail to the current and improved methodologies and rationalize the potential for improvement.

- Chapter 4, utility of experimental design: An example. This chapter presents our simulated data upon which we employ both the current and improved methodologies. We demonstrate the benefits of individual methodology components thereby demonstrating the potential for improvement.
- Chapter 5, sensitivity of benefits to variance in performance data. Since the benefits of each methodology depend on the nature of the simulated data, we vary our hypothetical data set and observe the change in the benefits. This approach verifies the robustness of the benefits of our improved methodology.
- Chapter 6, conclusions and recommendations. In the last chapter, we review our approach, summarize the results, theorize on the impact of improvements, and make general recommendations for the implementation of the improved methodology and further research in this area.

We meet our research objective by demonstrating a potential for improvement, presenting a sample of the impact of our improved methodology, and confirming the robustness of the improved method with respect to potential observed data sets. Figure 1.3 illustrates how these components are linked to the phases of testing.

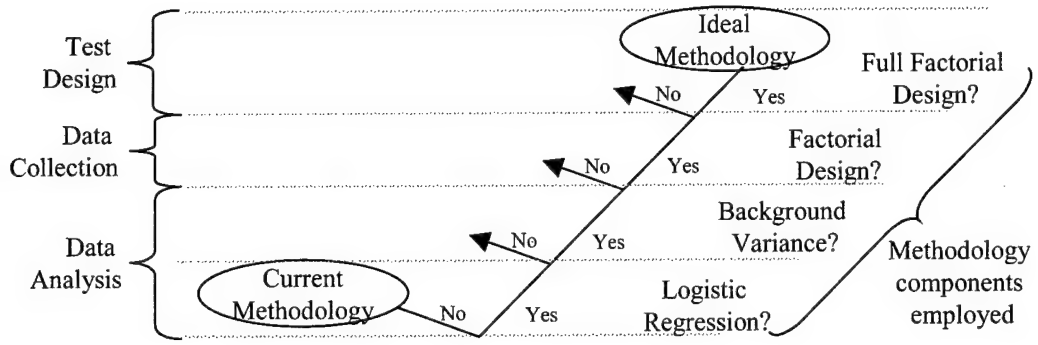

Figure 1.3 Relationship Between Methodology Components and Test Phases

Figure 1.3 shows four methodology components and their location in the test process with respect to the three phases. The tree diagram shows possible decisions faced by a test analyst when choosing a methodology. For instance, under the current methodology, we do not use any of the four components in the figure, but if we follow a series of decisions to implement these components (as we move upward, against the flow of the test phases) we arrive at an improved methodology. This methodology is ideal with respect to the methodology components (our recommended improvements). We use this decision structure to develop intermediate methodologies in chapter 5. In the next chapter, we describe automatic target recognition performance evaluation in greater detail.

2. REVIEW OF AUTOMATIC TARGET RECOGNITION PERFORMANCE EVALUATIONS.

In this chapter, we review processes pertinent to the field of ATR performance evaluation. We review these processes to explain, justify, and exhibit the feasibility of our recommendations. The topics covered in this chapter are below.

- Experimental design paradigm: A comprehensive methodology that encompasses techniques in all three test phases.
- Experiment building: The process and techniques of ATR test design.
- Image data: The object of our data collection effort.
- Measuring and reporting performance: Basic ATR performance measures common to all our methodologies.
- Analysis of proportion data: Techniques for analyzing our common performance measures.

We present support for our recommendations, but we do not address specifics until chapter 3. We begin our review with a discussion of experimental design.

2.1. Experimental Design Paradigm

We review the basic concepts of experimental design within each phase of testing. We do not address our recommended improvements until we have described current practices in ATR evaluation.

2.1.1. Test Design.

The experimental design paradigm has application to the design phase of testing for ATR evaluations. Using this paradigm, we identify factors that potentially have a significant effect on our performance, we control the factors that we can, and observe any uncontrollable factors. Using our control factors, we identify levels for each factor and construct a full factorial design (each condition is a unique set of factor levels where we have one condition for every possible combination of factors and levels, this is illustrated later). We detach a fraction of the design (using techniques called aliasing and blocking) so that we only collect the data we need to address test objectives. The resulting condition matrix is passed on to the next phase of testing.

2.1.2. Data Collection.

In data collection, we begin with a broad test design (a design where factors have few levels and the levels are near the extremes of the factor's possible settings) and collect the desired information for each condition in random order. The results are analyzed and we may return several times to the design phase to refine or add to our original design and repeat the collection phase. This design-collect-analyze cycle is known as iteration [6] (also part of the scientific method, see Figure 1.1). We remain in this cycle until we are satisfied that there is no sufficient benefit to continue data collection. In iteration, we maximize the efficiency of our data collection by first *screening* our factors to identify those that affect performance, then *characterizing* the effect of each significant factor, and finally, *confirming* the answers to our test objectives. Each of the latter two stages of iteration relies on the analysis results from the preceding stage.

2.1.3. Data Analysis.

After data collection, we typically use a technique called analysis of variance which uses the test factors as predictors and builds a model of performance. We use this technique to identify those factors and interactions between factors that explain the most variance in the performance measures.

This brief description of experimental design captures the types of recommendations we make in our research. The sections below describe some of the current practices in ATR evaluation and present our recommended improvements.

2.2. ATR Experiment Building (Part of Test Design)

The COMPASE Center at AFRL/SN defines experiment design as the binning and sequestration of previously collected images (for specific categories of evaluations) [8]. In contrast, experimental design (as recognized in the academic community) dictates the manner in which data is collected and analyzed, as well as the method of organization [6]. The current and improved design concepts we discuss here are one-at-a-time test design and factorial test design.

2.2.1. One-at-a-time Design Concept.

Air Force standard guidance for test and evaluation does not specify a specific method for test design with regard to experimental design [9; 10], (i.e., there is no handbook that dictates a method for producing test conditions in a design sense) and the ATR working group (ATRWG) data collection guidelines [3] do not explicitly address experiment design (though designed experimentation is recommended as a general approach in an earlier ATRWG document [4]). Test conditions in a typical data

collection effort are driven by the types of targets an algorithm of interest is designed to recognize and the context in which it should recognize them (environment, terrain, operational configuration, etc.). According to Ross [21], a common practice is one-at-a-time test condition variation. This practice involves beginning with a baseline condition (i.e., T-72 tank, on grass, turret forward) and collecting images at various aspect angles. Next, a single factor is altered (e.g.., turret rotated 30 degrees) and data is collected again. In the one-at-a-time method, the turret would be returned to the baseline position before any other factors are varied. The result is that complex combinations of test factors are seldom, if ever, tested.

2.2.2. Factorial Design Concept.

Factorial experimentation, or testing all possible combinations of a given set of controlled variables with finite levels, is useful in experimental design for estimating the nature of the effect of multiple variables on a response measure. A common complaint about factorial experimentation is that when many variables are involved, it is too costly or too complicated to test all combinations, and a one-at-a-time approach is preferred. However, in a technical report dated 1990 [4], the ATR working group (ATRWG) asserts that a factorial approach is a more efficient and effective means of experimentation than the latter option *especially* when data points are costly and many. Furthermore, factorial experimentation can be modified to accommodate resource limitations. Some examples are fractional factorial designs, blocking, and simple designs (e.g.: 2 levels per variable).

2.3. ATR Image Data Characterization (Part of Data Collection)

A sensor image is an electronic snapshot of an object of potential military interest. The image type is determined by the characteristics of the sensor used to collect the image. Image types can be radio frequency (RF), infrared (IR), or electro-optical (EO) in multiple bandwidths, creating the possibility for a wide variety of image types with characteristics in multiple electro-magnetic spectra. Algorithms are designed to take advantage of the unique characteristics (in a particular medium) of military targets and use the information (within an image containing an object of interest) to select a likely military system, based on a comparison to a known image database or on a model of target parameters. We do not address the specific media, hardware, or software used to collect images, rather we focus on the method used to collect images.

2.3.1. Image Data Accuracy.

Given that we have image data, to analyze the performance of a classification algorithm we need to know the truth about the imagery (i.e., to evaluate whether an algorithm has correctly recognized an object, we need to know with certainty what the object is). Accurate characterization of image data is essential to correct evaluation of algorithm performance since inaccuracies in truth data (true identification and location information to which algorithm results will be compared) bias evaluation results. Detailed image characterization presents a tedious task because much of the data must be hand-inspected to ensure quality truth data. Sims [22] points out that although much attention has been paid to evaluating algorithms, only a coarse characterization has been performed on the vast archives of image data. Due to the overwhelming difficulty of

characterizing existing data, information is often limited to target type, target state, and general environment data.

2.3.2. Image Data Coarseness.

Another difficulty is the coarseness of image characteristics data [11]. Target type must be known in order to evaluate whether correct identification has occurred. Location information is necessary in case several targets appear in one image. Other parameters such as target configuration, azimuth, aspect angle, and environment are desirable to explain more of the variance in an algorithm's performance. Additional characteristics exist that are not currently measured that could be used to further explain performance variance. Weszka [24] introduces several texture measures for classifying terrain. Target resolution can be measured directly by including the number of pixels in an image located directly on the target. Sims [22] demonstrates that the signal to clutter ratio provides a good indication of how an ATR algorithm will perform. If even a rough estimate of the signal to noise ratio could be included with an image, the potential exists to more precisely predict performance. Power [19] asserts that image quality can be a major determining factor in ATR performance and recommends (in addition to signal to noise ratio) human vision data to measure image quality. Due to the growing size of image data repositories and the need for accurate truth data, much work has been done to develop methods and software for quality assessment. Michel [16] et. al. recommend a statistical model for the automation of image quality assessment.

The point of this discussion is to show that the potential for improved image characterization exists and is well documented. Our review of images supports our

recommendation to increase the detail of image characterization by demonstrating the feasibility of our recommendation.

2.4. ATR Algorithm Performance Measurement and Reporting

We discuss the measurement and reporting of algorithm performance to familiarize the reader with the basic elements of an evaluation (dependent variables). Estimates of probability of detection, probability of identification given a detection, and probability of a false alarm are the primary measures of algorithm performance. Reporting performance is accomplished through various transformations of these measures. Confusion matrices are tables in which the target systems are listed along the horizontal and vertical axes, and the data in the table is the estimated probability of classifying a system as system A when the true identity is system B. Table 2.1 is a simplified example of a confusion matrix.

Table 2.1 Example Confusion Matrix for Two Systems

Percent Identification		Reported (%)		
		System A	System B	Other
Truth	System A	90	05	05
	System B	10	85	05
	Other	00	05	95

Confidence intervals about the mean probability of detection, identification, or false alarm are usually constructed assuming a binomial distribution for the number of occurrences of each event [2]. For example, let $\hat{p}$ represent the estimated probability for an event, and let n represent the number of opportunities for an event to occur, then

$$\hat{p} \pm Z_{(1-\frac{\alpha}{2})} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \quad \textbf{(2.1)}$$

is a (1-α)% confidence interval about the mean probability of the event, where the value $Z_{(1-\frac{\alpha}{2})}$ is a statistic that is used to generate intervals that would include the true probability with roughly a 95% success rate for $\alpha = .05$, assuming a normal approximation for the binomial distribution. For comparing the performance estimates from two algorithms, it is useful to estimate the difference between probabilities $(\hat{p}_2 - \hat{p}_1)$ and construct a confidence interval about the difference:

$$(\hat{p}_2 - \hat{p}_1) \pm Z_{(1-\frac{\alpha}{2})} \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}} \quad \textbf{(2.2)}$$

where n_1 and n_2 are the sample sizes (number of instances in which a positive identification, detection, or false alarm could have occurred) for each evaluation. If there exists a background variable (e.g.: azimuth from target to sensor) then the interval can be improved by pairing like angles and taking the difference between measures to generate a data set of paired differences. A new confidence interval for the difference between the probabilities becomes:

$$\frac{\sum_{i=1}^{n} (\hat{p}_{2i} - \hat{p}_{1i})}{n} \pm Z_{(1-\frac{\alpha}{2})} \frac{s}{\sqrt{n}} \quad \textbf{(2.3)}$$

where,

$$s = \sqrt{\frac{1}{n-1} \cdot \sum \left[(\hat{p}_{2i} - \hat{p}_{1i}) - \left(\frac{\sum_{i=1}^{n} (\hat{p}_{2i} - \hat{p}_{1i})}{n} \right) \right]^2} \quad \textbf{(2.4)}$$

The sample size is assumed to be equal for both populations and s is an estimate of the standard deviation of the paired differences. These confidence intervals can be used to perform simple hypothesis tests for the performance of two algorithms. The calculation in Equation 2.1 can be used to test the following hypothesis:

$$H_0 : p \leq p_a$$
$$H_a : p > p_a$$

(where p_a is a constant standard to which performance will be compared) by calculating the one-sided tolerance limit for $\hat{p}$:

$$\hat{p} + Z_{(1-\alpha)}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \qquad \textbf{(2.5)}$$

If the tolerance limit is greater than p_a, we reject the hypothesis that $p \leq p_a$ and conclude that $p > p_a$ (the value α in Equation 2.5 identifies our estimated probability of making an incorrect conclusion). Similar hypothesis tests can be constructed for the difference (or paired difference) between probabilities.

As identified in the confusion matrix (Table 2.1), there is both a probability for success and a probability for a false alarm for any system. We can estimate the change in detection probability as false alarm probability varies. A receiver operating characteristic (ROC) curve is used as a means to communicate the relationship between the probability of detection and the probability of false alarm. Given that algorithm performance is described by the example ROC curve in Figure 2.1, moving along the plotted line is a result of varying detection thresholds in the algorithm or varying degrees of clutter in the image data. For a review of the above techniques, see Alsing [1] or ATRWG 86-001 [5].

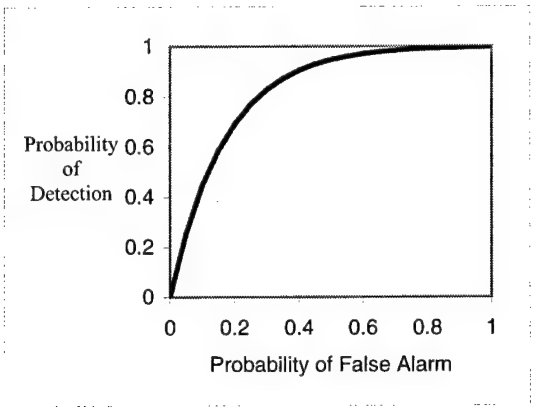

Figure 2.1 Example Receiver Operating Characteristic Curve for Probability of Detection

2.5. Analysis of Proportion Data (Part of Data Analysis)

We discuss the current analysis techniques and improved techniques in general and demonstrate the improved techniques are statistically valid and the current techniques imply many assumptions (e.g., we assume no interaction between factors). The topics we cover here are the standard performance model and the logistic response model. It is appropriate to treat the outcome of an attempt to detect, classify, or identify a target as a Bernoulli random variable since we classify an outcome as either a *success* or *failure*. There are many statistical methods focused on the analysis of rates and proportions (for examples, see Fliess [12]); for designed experiments, we will consider logistic regression (for a detailed explanation, see Neter, et. al. [18]).

2.5.1. Standard Model.

It is desirable to treat binary data as a special case for regression since two assumptions of the standard normal error regression model are necessarily violated. The standard regression model is:

$$Y_i = \beta_0 + \beta_1 \cdot X_i + \varepsilon_i \qquad \textbf{(2.6)}$$

where Y_i is the i^{th} outcome of i = 1 to n Bernoulli trials (in which only two outcomes are possible, 0 or 1), the β terms are the regression coefficients (chosen mathematically using a technique called maximum likelihood estimation), X_i is the setting of a prediction variable, and ε_i is our error, or naturally occurring randomness. The violations are:

1. *Normality of error terms*: Since each error term can only take on two values, $\varepsilon_i = 1 - \beta_0 - \beta_1 X_i$ when $Y_i = 1$, and $\varepsilon_i = -\beta_0 - \beta_1 X_i$ when $Y_i = 0$, the assumption that the ε_i are normally distributed is not appropriate.

2. *Constant error variance*: Since Y_i is a Bernoulli random variable with parameter π_i (representing the probability of observing a 1 in the Y variable), the variance for Y_i is $\pi_i(1-\pi_i)$. Also, because $\varepsilon_i = Y_i - \pi_i$, where π_i is a constant, we see the variance of ε_i is the same as the variance of Y_i. Substituting $\pi_i = E[Y_i] = \beta_0 + \beta_1 X_i$ (read: π_i is the expected value of Y_i) shows that ε_i is a function of X_i (see *normality of error terms*) hence the error variance depends on X_i and will differ for different levels of X.

In addition to assumption violations, another problem with binary data is the constraint it places upon the response function. Since the response function represents a probability, inferences about the mean response should be constrained by 0 and 1. A normal linear response function does not necessarily meet this constraint. Having pointed out these deficiencies, we propose the logistic regression technique as an alternative method without the same criticisms.

2.5.2. Logistic Model.

The basis of logistic regression is the Bernoulli probability mass function, which has the form:

$$f(x) = \pi^{Y} (1-\pi)^{1-Y} \tag{2.7}$$

where $Y \in \{0,1\}$. The logistic response function is of the form:

$$E[Y] = \frac{\exp(\beta_0 + \beta_1 X)}{1+\exp(\beta_0 + \beta_1 X)} \tag{2.8}$$

The estimates of the parameters $\beta_0, \beta_1, \ldots, \beta_i$ are determined using maximum likelihood estimation, but instead of the normal equations, we use the log of the logistic likelihood function:

$$\log_e L(\beta_0, \beta_1) = \sum_{i=1}^{n} Y_i(\beta_0 + \beta_1 X_i) - \sum_{i=1}^{n} \log_e[1+\exp(\beta_0 + \beta_1 X_i)] \tag{2.9}$$

The values for the coefficients that maximize this function can be found through numerical search procedures, available in some statistical software packages (in our research, we use JMP IN statistics, version 3.2.6, SAS Institute, Inc.). The use of the logistic response function relaxes the assumptions of normality and equal variance of the error terms and transforms the response to meet the boundary constraints, 0 and 1.

Because we do not violate the assumptions of our model, we believe that the logistic regression technique is more appropriate for our data than the current techniques.

Chapter 2 presents a general review of ATR evaluation. We briefly describe current test techniques and our recommended improvements. We also present support information to give the reader a clear picture of the ATR performance evaluation process. In chapter 3, we observe closely the specifics of current practices and clarify our recommendations within each test phase.

3. CURRENT PRACTICES IN THE EVALUATION CYCLE AND PROPOSED IMPROVEMENTS.

In this chapter, we scrutinize the current methodology and point out areas for potential improvement. The recommended improvements are explained in greater detail with simple examples. The chapter is organized into three sections (one for each test phase) and each section addresses the four issues below.

- Current methodology in ATR performance evaluations
- Potential areas for improvement in evaluation methodology
- Recommended improvements to evaluation methodology
- Potential benefits of improved methodology

These issues are addressed for each phase of testing, and the collection of recommendations comprises an improved methodology.

3.1. Phase 1: Test Design

The first phase of testing, test design, is critical because mistakes or poor decisions in this stage are usually irrecoverable. Selecting a design with which to collect data requires consideration of the test objectives, economical use of resources, and the multitude of conditions an operational system may face. The last consideration is included because our choice of test conditions, or rather, our decision not to include other conditions implies an underlying assumption that phenomena under the omitted conditions do not affect our ultimate test decision.

3.1.1. Current Design Methodology: One-at-a-time testing.

To illustrate the importance of test design, consider an example in which only three target factors are considered: turret articulation, camouflage, and revetments (target partially obscured by manmade objects). A one-at-a-time approach to test design results in at least four test conditions: a nominal case in which no factors are varied, and the three cases in which one of each of the three factors is varied. It is also possible that we wish to test different levels of each factor, such as 10, 30, and 45 degrees for turret articulation, or different types of camouflage. If we treat all cases when the same factor is being varied as one condition (e.g., 0 degrees of turret articulation falls under condition one, and 10, 30, or 45 degrees of articulation falls under condition two), the resulting matrix is shown in Table 3.1.

Table 3.1 One-at-a-time Test Conditions for Three Factors

Condition	Turret Articulation	Camouflage	Revetments
1	No	No	No
2	Yes	No	No
3	No	Yes	No
4	No	No	Yes

3.1.2. Areas for Improvement in Test Design Methodology.

One might assume the design in Table 3.1 is most efficient for collecting information about the effect these variables have on ATR performance. The first criticism of this design is that it fails to capture information about ATR performance when two or more factors are varied simultaneously. Second, for any factor we have

three observations of performance when one factor is not varied, but only one observation when the factor is varied, resulting in imbalanced data. Imbalanced data becomes a problem when we wish to compute confidence intervals or perform hypothesis testing, as we will demonstrate. For these reasons, the design may frequently fall short of our objective to provide accurate results.

3.1.3. Recommended Improvements: Factorial Testing and Fractionation.

We recommend factorial design and fractionation to improve our test methodology. We would like to minimize uncertainty in our test, but the one-at-a-time design provides no knowledge of the four possibilities in which more than one factor is varied simultaneously, which we will refer to as two and three-factor interaction effects. Table 3.2 shows the full matrix of possible conditions in which for every level of one factor, we collect data on all levels of the other factors, also known as a factorial design. We should recall here that within each condition, the factor being varied can take on multiple levels. If we are interested in a detailed characterization of performance across the multiple levels, the conditions in the design in Table 3.2 increase. An obvious objection to factorial testing is that with many test factors, collecting all of the combinations may be infeasible. For example, if there are seven factors being considered, even with only two levels (varied and not varied) there are $2^7 = 128$ possible test conditions. In this case it is tempting to adopt the one-at-a-time approach which requires only 8 conditions, allowing us to consider expanding the design to include more levels per factor.

Table 3.2 Full Factorial Conditions for Three Factors

Condition	Turret Articulation	Camouflage	Revetments
1	No	No	No
2	Yes	No	No
3	No	Yes	No
4*	Yes	Yes	No
5	No	No	Yes
6*	Yes	No	Yes
7*	No	Yes	Yes
8**	Yes	Yes	Yes
Note: * two-factor interaction, ** three-factor interaction			

Fortunately, it is not necessary collect all 128 conditions for the full design. The full design allows us to estimate the effect of every combination of multiple factors (including the effect of varying all seven factors at once). We do not expect all these effects to each be significant. In fact, since our response is bounded, once we have degraded performance effectively to zero, varying more factors cannot significantly degrade performance. To take advantage of this knowledge we utilize the concept of fractionation.

Consider again the original example with only three factors. If we are limited to only four test conditions, we can choose those conditions that allow us to extract the maximum information about the entire set of possible conditions. The information in Table 3.3 represents a half-fraction of the full collection design. If we only collect those runs that are not shaded, we can still estimate the effects of single factors.

3.1.4. Potential Benefits of Factorial Design and Fractionation.

By designing factorial experiments, we ensure that we will be able to estimate our performance under every possible combination of these factors. Another benefit is that for each factor, we collect four observations for every level.

Table 3.3 Half-fraction of Full Factorial Conditions for Three Factors

Condition	Turret Articulation	Camouflage	Revetments
SOC	No	No	No
	Yes	No	No
	No	Yes	No
Turret & Camo	Yes	Yes	No
	No	No	Yes
Turret & Revet	Yes	No	Yes
Camo & Revet	No	Yes	Yes
	Yes	Yes	Yes

Finally, all factors are orthogonal in the design matrix (i.e., for each factor, we collect observations at both levels for every combination of the other factors). Figure 3.1 illustrates these characteristics.

There are twice as many conditions for the factorial experiment so we fractionate to manage the test design. The benefit of fractionating our factorial designs is that we can reduce our condition matrix, and as we increase the number of factors we want to estimate, we also increase the number of estimable interactions. Figure 3.2 shows the full factorial and fractional factorial designs. The three factor design may not be the best case for demonstrating the utility of fractionating designs.

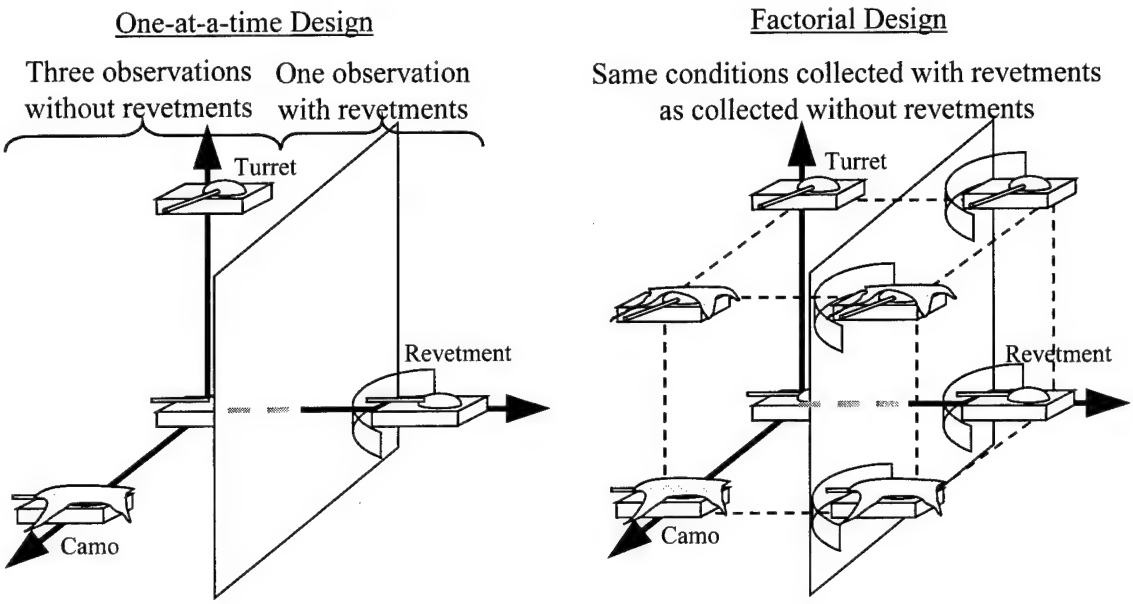

Figure 3.1 Comparison of One-at-a-time and Factorial Conditions in Three Factors

Consider a test in which we want to collect performance data on seven factors. Recall that with seven factors, each with two levels, there are 128 possible combinations. If we select conditions using a technique called fractionation (resulting in some higher order multiple effects becoming "aliased" with other effects, and inestimable), we can elect to run only 16 of the 128 total conditions (a $1/8^{th}$ fraction) and still estimate the effects of all seven factors and their two-factor interactions. By using fractional designs we give up the ability to distinguish between many lower and higher order factor effects. We accept this loss because we do not expect complex high order interactions to have a significant effect on performance. With this in mind, we fractionate large designs to either reduce the size of infeasible condition matrices, or gather more information in the same number of collected conditions.

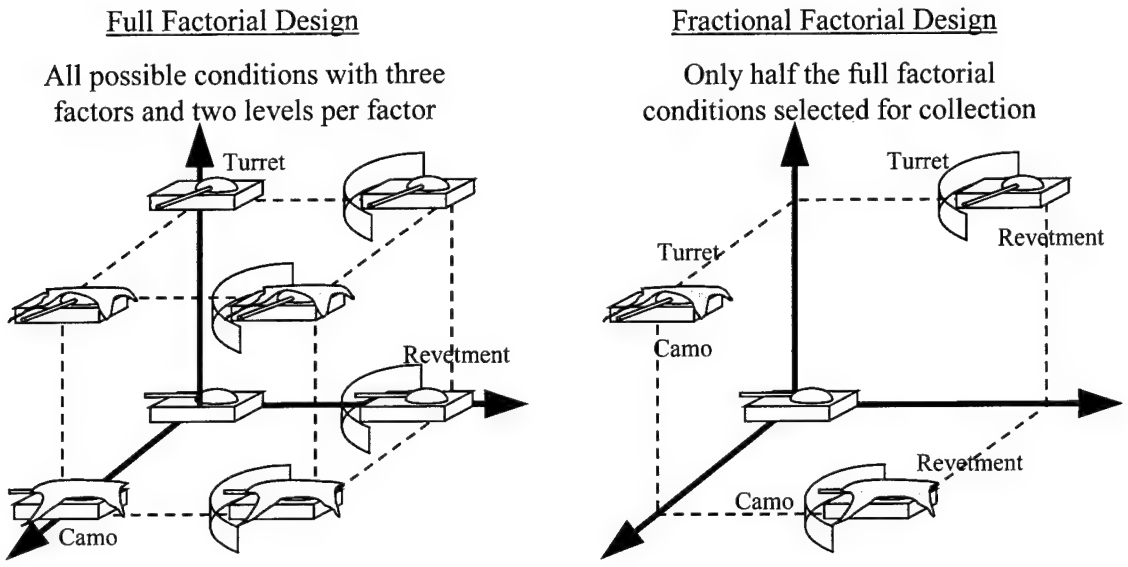

Figure 3.2 Comparison of Full and Fractional Factorial Conditions in Three Factors

3.2. Phase 2: Data Collection

Oversights in the second phase of testing, data collection, are less serious, but recovery is often prohibitively difficult. Efficient data collection includes measuring any factors (controlled or otherwise) that may affect our dependent variable, and allowing our overall test design to react to unforeseen phenomena in the data. Coarse measurement of background factors and inflexible design imply an assumption that our current knowledge of the test outcome is sufficient to reject the possibility of phenomena more complex than our results can be used to estimate.

3.2.1. Current Collection Methodology: Coarse Data and One Shot Collection.

Coarse data characterization, as we identified in chapter 2, is a fact in ATR performance evaluations. Another practice is waiting until all the data has been collected

before performing any analysis. Recall, in Figure 1.1 the scientific method includes a provision to return to the design phase and refine the focus of our data collection. Without planning this recursion into our test (even if we do reserve some resources to explore in our experiment) our test collection methodology is effectively a one shot effort. By delaying (or not developing the capability to analyze data in a short enough time frame), we sacrifice many opportunities.

3.2.2. Potential for Improvement in Data Collection Methodology.

Before collecting any data, it is important to consider what level of detail we want to describe our data. If the only information we record during collection is a basic description of the scenario, we lose the ability to enter other factors into our analysis. The cost of not knowing the state of factors that effect our performance is an inflated estimate of our background variance, which limits the power of hypothesis tests for the significance of our analyzed factors. Suppose the collected data contains anomalies that should be further explored, or some data is corrupted and cannot be used; these incidents might not be addressed in time to take action unless the data is examined immediately. These are examples of how coarse characterization and one shot collection can cost us opportunities to make our experiment more accurate and efficient.

3.2.3. Recommended Improvements: Iteration and Detailed Characterization.

We recommend a more detailed characterization of image data. In our example, we identify eight conditions which are determined by three variables. Recall that for each condition we collect images at different angles, (one for every nine degrees of azimuth in our example). If we record the exact azimuth we can enter this factor as

another variable in our analysis. Some factors may be too costly or difficult (dielectric coefficient for each square foot of background area) or insignificant (wind velocity), yet many relevant factors would be recorded if we identified them ahead of time as potentially important variables.

We also recommend an iterative approach to data collection. Assume we have the ability to assess algorithm performance immediately after the collection of each successive sensor image. We can let our recent estimates of performance influence the collection of new data (to maximize our knowledge of the total space of interesting conditions). We can either expand or narrow the scope of our collection as appropriate to best answer our test objectives.

3.2.4. Potential Benefits of Iteration and Detailed Characterization.

The benefit of iteration lies in the potential to preserve our resources for exploring the most significant test phenomena. If we discover that we have consistent performance in one condition, we can focus on test conditions where performance varies widely. If we find that a factor does not affect our performance, we can neglect to vary that factor in the remaining collection effort. An iterative collection sequence is a fundamental component of DOE [6] and it empowers us to focus our collection where it gains us the most knowledge. We use screening, characterization, and confirmation as the basic steps in an iterative approach. In the screening stage of testing, we can use our half fraction design to determine which factors (if any) have a significant effect on our performance. In the characterization stage, we use a better design (perhaps with more levels per factor or several repetitions) with the remaining factors to estimate multiple factor effects. In the

confirmation stage, we augment our design, for instance, to add confidence to conditions where we observe inconsistent performance.

The benefit of detailed data characterization is evident if we recall the impact of not gathering information on a factor that affects performance. If an underlying factor causes performance to either improve or degrade as it varies, these fluctuations in our performance measures go unexplained. Unexplained variance becomes our estimate of noise in the data. When our noise estimate is inflated due to unmeasured factors, the result is lower confidence in our results, or less precision in our confidence intervals.

3.3. Phase 3: Data Analysis

Our data analysis methodology has no direct impact on design or collection, but there is still the possibility of making poor decisions with good data by implementing ineffective analysis techniques. A good analysis technique should lead us to understand those relationships among our test variables that are relevant to our test objective. An appropriate technique should be selected for its effectiveness in identifying answers to objectives and the appropriateness of the technique assumptions.

3.3.1. Current Analysis Methodology: Brute Force and Normal Error.

If we generate tables, statistics of location and scale, and scatterplots of data to answer test objectives, we are performing analysis solely by brute force (as opposed to using advanced analysis techniques to explore the data, and only plotting relationships that we know will be interesting). All methods for investigation can be classified as analysis, but with brute force, we fail to avail ourselves of efficient techniques to reduce our data and identify key relationships. Also, recall that in chapter 2 we revealed that our

interval estimation was accomplished using a standard normal model, or a normal approximation to the binomial distribution which is inaccurate for binary data.

3.3.2. Potential for Improvement in Data Analysis Methodology.

Analysis by brute force is cumbersome and is not guaranteed to reveal important relationships in the data. The multitude of possible graphs and tables an analyst must peruse to discover complex relationships is overwhelming. Time constraints, abundance of data, or even ignorance of its existence may cause us to overlook information that is relevant to our test objectives. In addition, the use of a standard normal regression model (the assumptions of which are necessarily violated with performance data) renders our confidence intervals suspect and potentially invalid. We may generate intervals that are unrealistically small and even fail to cover the true performance parameter we wish to estimate. An analysis methodology should guide us to relevant results and the assumptions should be appropriate for our data.

3.3.3. Recommended Improvement: Logistic Regression.

Logistic regression accepts our test factors and performance measures as inputs and produces coefficients that are used to construct a model of performance. Unlike linear regression, the coefficients (β's) in logistic regression do not represent the change in the response for a unit increase in a predictor. Since the variance of our response depends on the level of the predictors, the magnitude of the effect of a predictor also depends on the location of the mean probability. To interpret a logistic regression coefficient (β), we define the odds of detection for a given condition, with detection probability p, to be:

$$odds = \frac{p}{1-p} \tag{3.1}$$

The odds for a condition are multiplied by e^{β} for every unit increase in the factor associated with β (for multiple factors, the odds are multiplied sequentially by e^{β_i} for each unit increase in x_i, for i=1 to p, where p is the number of factors). For example, if our detection probability is 0.50 when factor 1 = 0, then the odds of detection are 1.0, or a 1 in 2 chance of detection. If β=1.1, then $e^{\beta} = 3$ and the odds increase from 1.0 to 3.0 (3 in 4 chance of detection), which is associated with a detection probability of 0.75. In other words, the increase in factor 1 triples our odds of detection. Note that if our initial detection probability is 0.10, the odds are 0.11 (0.11 in 1.11) and factor 1 triples the odd to 0.33 (0.33 in 1.33) or a 0.25 detection probability, so the increase depends on the starting point. Also, since we code our factor levels as –1 and 1 for low and high settings in experimental design, a change from one level to the next is actually two units in our coded scale. The impact of this convention is that to estimate our factor effects from the coefficients, we need to double the increase in the odds.

To make inferences about the coefficients, mean performance estimates and new predicted observations, we must estimate the variance-covariance matrix of the predictor variables. The matrix is formed by first generating the Hessian matrix from the log-likelihood function [18]. The entry in the i, j^{th} cell is the second derivative of the log likelihood function with respect to β_i, β_j. The variance-covariance matrix is the inverse of the negative Hessian matrix (taking the negative of all entries). The variance-covariance matrix is represented by $s^2(\underline{b})$, where $\underline{b}$ is a matrix containing the parameters we estimate with the β's. To calculate confidence intervals about the value of the k^{th} coefficient we use:

$$b_k \pm z(1-\alpha/2)s(b)_{k,k} \quad (3.2)$$

To calculate simultaneous intervals for g coefficients, for each coefficient use:

$$b_k \pm B \cdot s(b)_{k,k} \quad (3.3)$$

where $B = z(1-\alpha/2g)$. Hypothesis testing for specific effects can be accomplished by evaluating whether the interval contains zero. To estimate the intervals about the mean response at one setting of the predictor variables, let X_h be the vector of values for the setting of interest, then use:

$$\frac{1}{1+\exp\left(-\beta' X_h \mp z(1-\alpha/2)\cdot\sqrt{X_h' s^2(b) X_h}\right)} \quad (3.4)$$

Predicted observations at a setting of a predictor variable are simply generated by evaluating the mean response against a classification rule (e.g., if the expected response at a condition is 0.6, then a " > 0.5 " classification rule would result in a prediction of "1", or, "success" for this case).

Hypothesis tests for entire models, goodness of fit, and residuals can also be accomplished using various techniques. One hypothesis test is derived from a statistic called the model deviance (DEV). The deviance of a regression model is defined to be the difference between the log-likelihood functions using the regression coefficients in place of the β's for the first function, and Y_i in place of β′X in the second function. To test whether two models are equivalent, we can calculate the deviance for each and the difference follows a chi-square distribution with p-q degrees of freedom (p predictors in the full model, q predictors in the reduced model). The hypothesis test goes as follows:

$$H_0 : \beta_i, \beta_{i+1}, \beta_{i+2}, \cdots \beta_j = 0$$
$$H_a : \text{Not } \forall \beta = 0$$

(where the β's in H_0 correspond to those omitted from the reduced model), and:

If $DEV_{reduced} - DEV_{full} > \chi^2(1-\alpha, p-q)$ then reject H_0 . This procedure is called a partial deviance test. To test for goodness of fit, evaluate $DEV_{reduced} > \chi^2(1-\alpha, n-q)$; if this inequality is true, conclude the model is a good fit. Other tests are derived from the chi-squared distribution, and F distribution. All these techniques are complicated compared to brute force investigation, but the benefits make the effort worthwhile.

3.3.4. Potential Benefits of Logistic Regression.

As we have demonstrated, the logistic response function is intended to estimate a binary response (such as detect/no detect). The assumptions previously violated are met with logistic regression and our approach has statistical rigor. Also, the regression technique is flexible and powerful, leading the analyst directly to key relationships in the data through coefficient magnitude and significance. The impact of not using such an elegant technique is potentially incomplete or misleading results, and tedious data investigation, effecting poor or late decisions.

We believe our assessment of the current methodology and our recommended improvements are compelling, but need to be demonstrated. We assert that the potential for improvement exists, and in the next chapter, develop a scenario that demonstrates this potential.

4. UTILITY OF DESIGNED EXPERIMENTS: AN EXAMPLE.

In this chapter, we use a simulated dataset to quantify the benefits of an improved methodology for one possible set of data. The dataset is simulated so that we can know the underlying population parameters and contrast the results from two methodologies. This discussion is organized in the following manner.

- General discussion of approach
- Method of data simulation
- Application of methodologies in test phases

Our intent in this chapter is to establish that our potential for improvement can be realized in a simplified scenario that is typical of an ATR evaluation.

4.1. Approach

Our basic approach is to build our improved methodology by sequentially adding recommended improvements (methodology components) one at a time, beginning with the simplest improvements. Improvements in the analysis phase of testing are the easiest, then collection, and finally the most difficult improvements occur in the test design phase.

We assert here that if we have the capability to implement difficult changes to our methodology, it makes sense to also implement simpler changes. This is because our changes in later phases of test design take advantage of changes in earlier phases. For example, if we design a factorial experiment for data collection, we also use an advanced analysis technique (like analysis of variance or regression) to analyze the data. For this

reason, we do not implement improvements in the design and collection phase without also improving the analysis phase. Similarly, we do not implement design improvements without collection improvements. The result is that our component-wise addition of recommended improvements moves backwards in the evaluation process. This approach has the advantage of demonstrating how our benefits increase as we improve our methodology further back in the test process. We identify four main improvements in this section that are used to develop five distinct methodologies. Figure 4.1 illustrates the relationship between the methodologies, methodology components, phases of testing and timeline.

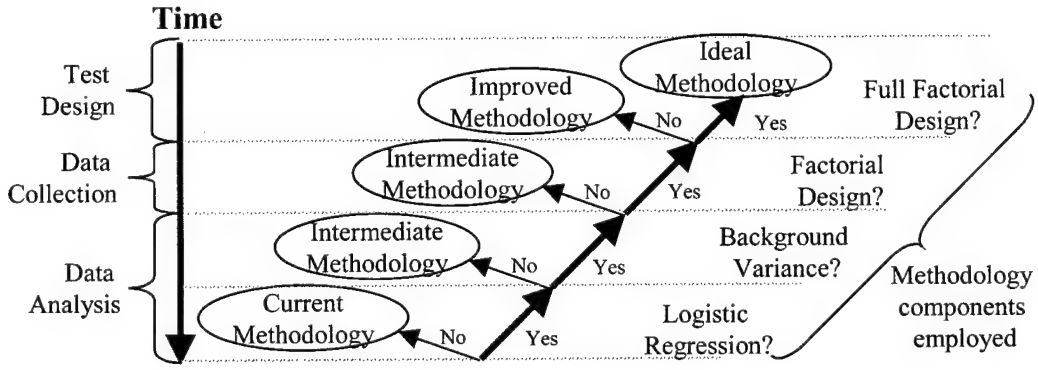

Figure 4.1 Relationship Between Test Timeline and Improved Methodologies

4.2. Simulating Performance Data

Suppose we wish to collect data to evaluate the effect of turret articulation, camouflage and revetments on the performance of two competing ATR algorithms. Recall, there are eight possible conditions (see Figure 3.1 and Table 3.2) in these three

variables when we treat any variation in a factor as a high level and no variation as a low level. Suppose also that we can know with certainty the precise probability of detection of our target in each of these eight configurations, with adjustments for different values of background factors. If we use a logistic response model for performance, we have a mean response (determined in regression by the intercept term) and an effect for each factor.

4.2.1. Factor Effect Coefficients (the Truth Model).

The effect of a factor on performance is denoted by an effect coefficient that is an input to the logistic response function. If we let (-1) denote a condition where factor A is not varied and (1) denote a condition where factor A is varied, then Figure 4.2 illustrates the relationship between factor levels, effect coefficients, and performance using a logistic response model. In this form, a set of coefficients (an intercept coefficient and one additional coefficient per factor) determines the performance for every condition. Since noise exists in our observed performance (denoted by curves in the figure), we expect our performance estimates to converge to the known performance parameters, based on our model. Two hypothetical sets of coefficients for competing algorithms are shown in Table 4.1. We use these coefficients to generate simulated data.

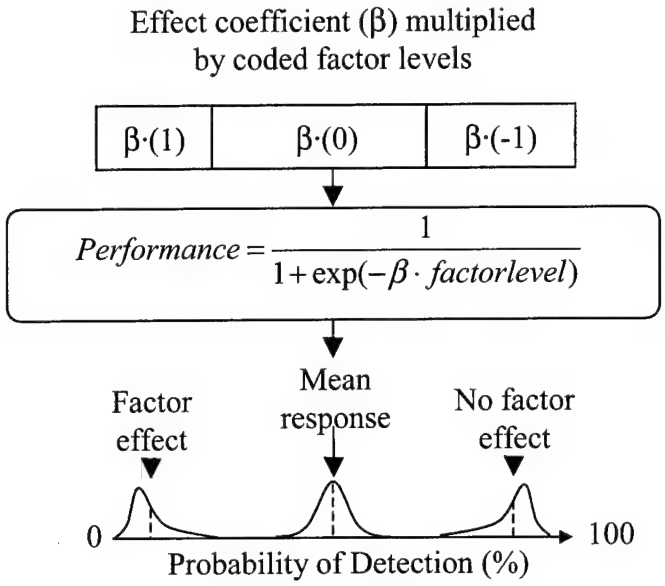

Figure 4.2 Relationship Between Coefficients and Response in Logistic Regression

Table 4.1 Hypothetical Factor Effect Coefficients for a Logistic Response

Coefficients	Mean Response	Turret	Camouflage	Revetments
Algorithm 1	1.2	-0.5	-0.6	-0.7
Algorithm 2	0.5	-0.3	-0.4	-0.5

In addition, we include coefficients that further perturb performance by creating an effect for an unknown background factor and all multiple factor effects. If two factors interact, the effect of one factor depends on the level of the other factor. Including multiple effects allows the degrade due to two factors to have a magnitude greater than the sum of the degrades due to each individual factor. The coefficients for each additional factor are in Table 4.2.

Table 4.2 Hypothetical Multiple Factor Effect Coefficients for a Logistic Response

Coefficients	Unknown Background Factor	Turret/ Camo Effect	Turret/ Revetment Effect	Camo/ Revetment Effect	Turret/ Camo/ Revetment
Algorithm 1	-0.01	-0.1	-0.2	-0.3	0
Algorithm 2	-0.01	0	-0.05	-0.05	0.1

4.2.2. Performance Calculations (Simulating Observations).

Using coefficients as inputs to the logistic response function (for each algorithm), we can calculate performance by varying the factor level settings (Equation 2.8). Table 4.3 contains the results from these calculations. These probabilities are our known parameters that we estimate in ATR performance evaluations.

Table 4.3 Calculated Detection Probabilities Using Hypothetical Effect Coefficients

Factor Level			Probability of Detection	
Turret	Camouflage	Revetments	Algorithm 1	Algorithm 2
-1	-1	-1	0.90	0.78
1	-1	-1	0.85	0.73
-1	1	-1	0.85	0.68
1	1	-1	0.85	0.66
-1	-1	1	0.73	0.52
1	-1	1	0.64	0.44
-1	1	1	0.54	0.40
1	1	1	0.19	0.28

We use our known parameters to generate random data for each condition, and across all levels of our background factor. Error is introduced into the dataset by generating random observations from several Bernoulli random variables with the same parameters as our known population (see Law et. al. [14]). To generate a random observation from a

Bernoulli distribution with parameter p, we first generate a random uniform number between 0 and 1. If the random number is less than or equal to our parameter p, our observation is classified as a success (detection), otherwise it is a failure.

4.3. Results Using Current Methodology

Recall that if we use a one-at-a-time approach, our design matrix is shown in Table 3.1. With four test conditions, we collect images from all aspect angles (one image every nine degrees) around the target of interest. This yields 40 images per condition, a total of 160 images for algorithm evaluation. After data collection, we have one observation (either *detected* or *not detected*) for each image and algorithm (320 total observations). Using our simulated data set, there are many tables and graphs we can generate to explore the performance of the two algorithms. Table 4.4 shows the mean performance for each condition (translated into a degrade from the baseline condition).

Table 4.4 Mean Detection Probabilities Using Simulated Data (One-at-a-time Conditions)

Algorithm Performance	Probability of Detection (%)	
	Algorithm 1	Algorithm 2
SOC	0.98	0.85
Turret	-0.10 (.88)	-0.00 (.85)
Camouflage	-0.18 (.80)	-0.10 (.75)
Revetment	-0.20 (.78)	-0.15 (.70)
Notes: SOC = Standard Operating Condition Last three rows represent delta percent off SOC		

These performance estimates do not match exactly the known performance parameters due to the randomness we have inserted in the data. We see algorithm 1 performs better than algorithm 2 in all conditions. Furthermore, we see that including revetments induces

the greatest degrade in our detection capability. Using the confidence intervals defined in Equation 2.1, we generate confidence intervals about the mean probability of detection for each of the two algorithms, as shown in Figure 4.3.

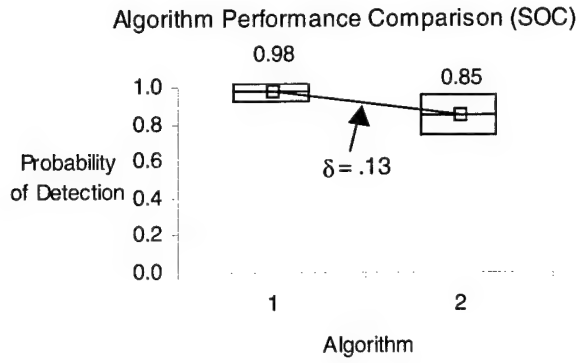

Figure 4.3 Confidence Intervals for Algorithm Performance, (Standard Operating Condition)

Using this method, the overlap of the two intervals indicates there may be insufficient statistical evidence to conclude the mean probability of detection for algorithm 1 exceeds the mean for algorithm 2. Figure 4.4 shows confidence intervals for the three single factor conditions.

The following discussion addresses three concerns with this methodology. The first two are minor issues of statistical rigor, and the last is a concern of efficiency. First, the method for generating confidence intervals leaves open the possibility of intervals that exceed the [0, 1] boundaries of our response measure, as shown in Table 4.5.

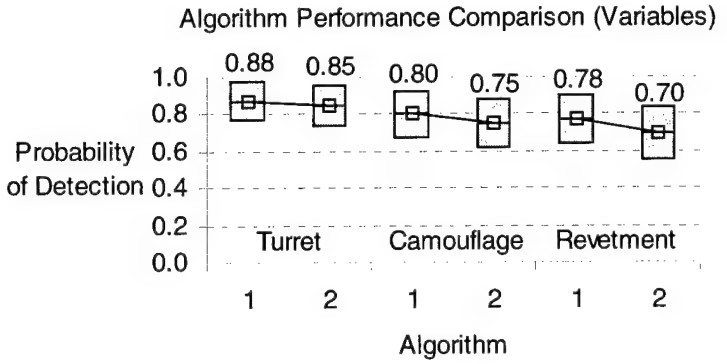

Figure 4.4 **Confidence Intervals for Algorithm Performance, Single Factor Conditions**

Table 4.5 **Upper and Lower Confidence Limits for One-at-a-time Conditions**

Confidence Intervals	Algorithm 1		Algorithm 2	
	lower confidence	upper confidence	lower confidence	upper confidence
SOC	0.93	**1.02**	0.74	0.96
Turret	0.77	0.98	0.74	0.96
Camouflage	0.68	0.92	0.62	0.88
Revetment	0.65	0.90	0.56	0.84

Traditionally, when the intervals exceed their boundaries, the analyst truncates the interval so that it stops at 0 or 1, as appropriate. Second, the lack of any hypothesis test omits the possibility of answering test objectives directly. For example, if the objective is: "Determine which algorithm performs better in the SOC case", an indirect answer might be: "The mean of algorithm 1 is greater than algorithm 2 for this sample data", caveated with: "there is overlap in the 95% confidence intervals". A direct (and therefore more desirable) answer is: "We are 95% confident that the mean of algorithm one is greater than algorithm two by at least 'δ' percentage points", where δ can be specified beforehand. The only mathematical difference between these two answers is

that the second relies on paired differences (see Equation 2.3) but this slightly modified approach allows more powerful statements. Our concerns regarding interval boundaries and hypotheses are easily remedied by taking advantage of the fact that we know our data comes from the binomial class of distributions, and using the logistic response function.

The last concern is over the cumbersome task of relying on exploration of graphs to identify important relationships in the data. Our example only includes one target, three factors, two algorithms, and assumes we are not interested in confusors (false targets), environment, background, etc. If we are tasked with comparing several algorithms across 20 targets, 50 factors, and multiple backgrounds and environments including confusors, we need to observe hundreds of tables and graphs just to make simple conclusions. Even in our simple example, we could make dozens of other comparisons and draw more conclusions. It is more convenient to use a tool that can help us identify interesting phenomena in a more efficient manner (and guarantee that the rest of the data contains no interesting information).

4.4. Implementation of Improvements in Phase 3: Data Analysis

Here, we will apply both the standard and proposed analysis methodologies to a simulated data set and observe the differences between the two sets of results. We will demonstrate that logistic regression and hypothesis testing is a preferable analysis approach even with a simple, small test.

4.4.1. Results Using Logistic Regression

We have already described the technique called logistic regression at length, so we now provide results from performing a logistic regression on the data analyzed

previously by brute force. Figure 4.5 illustrates the basic process in logistic regression and shows that the output becomes the input for the logistic response function.

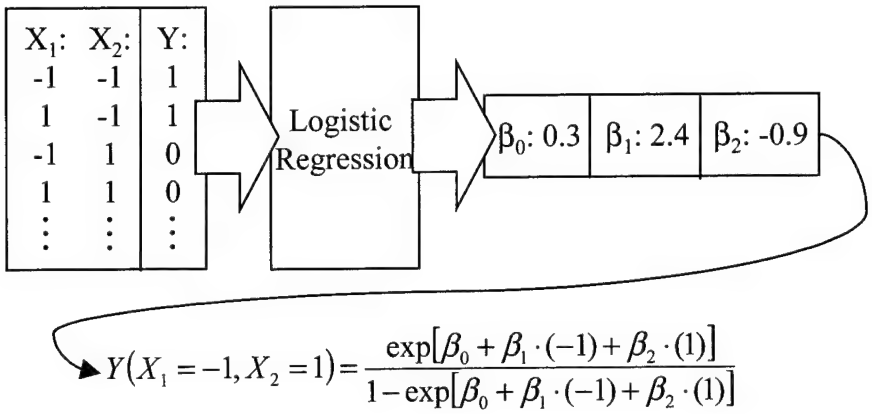

Figure 4.5 Logistic Regression Process for Binomial Response Data

Given our inputs, (independent variables and binary dependent variable) the technique provides us with coefficients that allow us to estimate the probability of observing a "1" in the response variable, given the settings of the predictor variables. Using the three factors as predictors in a logistic regression model (coded appropriately) results in Table 4.6. The values in the estimate column are the β's in our model, analogous to linear regression coefficients in that we multiply them by the coded variables, but the mean response is calculated via the logistic response function (Equation 2.8). We are also provided with the standard error of the β coefficients so that we can perform hypothesis tests on the significance of factor effects (p-value columns).

Table 4.6 Logistic Regression Output, Three Factors, One-at-a-time Design

Logistic Regression Results	Algorithm One			Algorithm Two		
	Estimate	Standard Error	p-value (χ^2 test)	Estimate	Standard Error	p-value (χ^2 test)
Intercept	0.453	0.623	**0.468**	0.973	0.401	**0.015**
Turret	-0.859	0.560	**0.125**	0.000	0.313	**1.000**
Camouflage	-1.139	0.544	**0.036**	-0.318	0.287	**0.268**
Revetment	-1.213	0.541	**0.025**	-0.444	0.281	**0.114**
Note: If p-value < 0.05, we reject the null hypothesis (b = 0) and accept the value in the estimate column as a significant effect.						

The standard error column is the standard deviation of the value in the estimate column. The p-value is the probability of obtaining the value in the estimate column (or greater) when the true value we are estimating is actually zero, based on a Wald chi-squared test. At a glance, only camouflage and revetments under algorithm 1 and the intercept term under algorithm 2 are greater than two standard errors from 0. Not coincidentally, if we had used a threshold of "<5%" to identify which effects were certainly not zero based on the p-value, we would select the same effects we selected before. If we use the logistic response function (Figure 4.5) with the coefficients in Table 4.6, we obtain the performance estimates in Table 4.7.

Table 4.7 Estimated Performance Using Logistic Response (One-at-a-time Conditions)

Algorithm Performance	Probability of Detection (%)	
	Algorithm 1	Algorithm 2
SOC	0.98	0.85
Turret	0.88	0.85
Camouflage	0.80	0.75
Revetment	0.78	0.70
Notes: SOC = Standard Operating Condition		

It is interesting that our calculations of the mean response at each of the four conditions match the averages from Table 4.4. In our evaluation, we wish to test the algorithm effect, or whether preferring one algorithm over the other affects our performance. We can add an algorithm factor to the analysis by including another dummy variable to represent algorithm 1 and 2. We code the algorithm factor [–1, 1]. Traditionally, we code dummy variables [0,1]; here, the p-values for the algorithm factor and the algorithm interactions do not differ for the two coding schemes. The results of this regression are in Table 4.8.

Table 4.8 Logistic Regression Output (p-values only), Three Factors Plus Algorithm Effect

Effect	p-value
Turret	0.181
Camouflage	**0.018**
Revetment	**0.007**
Algorithm	0.483
Turret*Algorithm	0.181
Camo*Algorithm	0.182
Revet*Algorithm	0.206

In this regression, we include three additional inputs whose values are the product of the coded values of each original factor and the algorithm factor. Including these as separate factors allows us to test whether an interaction exists between the two factors (we say the variables interact if the effect of each factor depends on the setting of the other factor). Based on the p-values in Table 4.8, we have evidence to conclude that the camouflage and revetment effects are not zero and are therefore statistically significant. If we trust this technique, we believe it is acceptable to graph these effects and ignore the rest since they are statistically insignificant. Figure 4.6 shows the percent degrade in performance

when each of our significant factors is varied (with confidence intervals based on Equation 2.2).

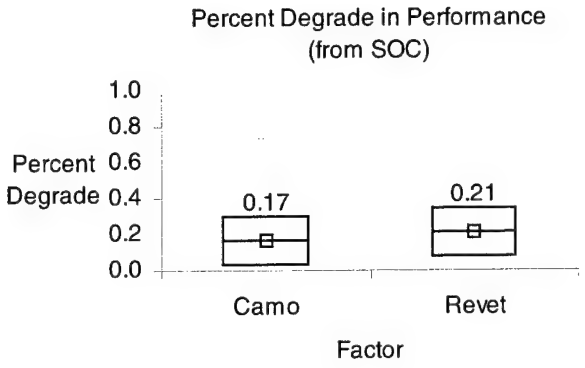

Figure 4.6 Performance Degrade from SOC (Due to Camouflage and Revetments)

To illustrate why no other effects are significant, consider the algorithm effect and its related interactions. If we break the camouflage and revetment effects up into algorithm one and two, we see similar results for both algorithms (see Figure 4.7).

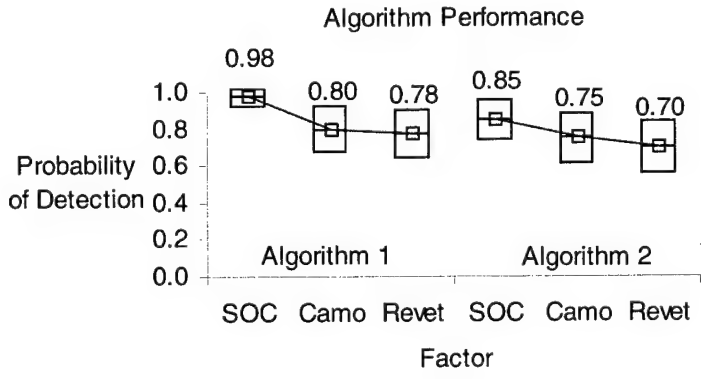

Figure 4.7 Performance Degrade Due to Camouflage and Revetments (Algorithms Separated)

Not only is there overlap for like conditions, but since the interactions are insignificant, the trend in each graph is the same. Figure 4.8 shows the effects with both algorithms combined.

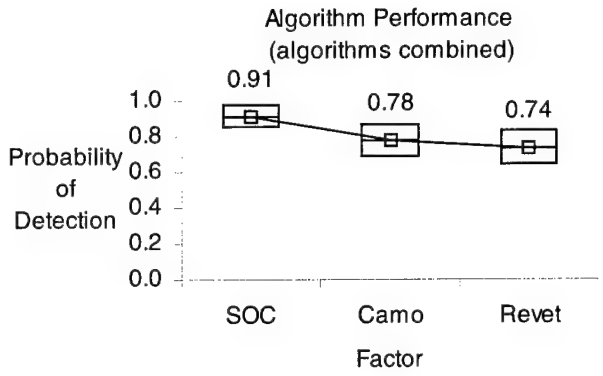

Figure 4.8 Performance Degrade Due to Camouflage and Revetments (Algorithms Combined)

Comparing Figures 4.7 and 4.8 shows that displaying results for both algorithms yields little additional information than a graph which combines the results, as we expected. Using the effect test to identify the types of graphs or tables that are most interesting is an efficient and effective means to analyze data.

4.4.2. Results Using Improved Confidence Intervals.

In all our graphs, we are using the normal approximation to the binomial distribution to generate confidence intervals and we have not stated how we might use hypothesis testing to answer test objectives. If we use the intervals defined in Equation 2.10 (replacing $z_{(1-\alpha/2)}$ with $z_{(1-\alpha/2g)}$ where g is the number of intervals we are generating),

then the Table 4.9 contains the new confidence intervals for the same conditions in Table 4.5.

Table 4.9 Confidence Intervals and Mean Response Using Logistic Regression

Logistic	Algorithm 1			Algorithm 2		
Response C.I.'s	Lower	Mean	Upper	Lower	Mean	Upper
SOC	0.71	0.98	1.00	0.63	0.85	0.95
Turret	0.66	0.88	0.96	0.63	0.85	0.95
Camouflage	0.58	0.80	0.92	0.53	0.75	0.89
Revetments	0.55	0.78	0.91	0.48	0.70	0.86

The intervals in Table 4.9 are wider than the intervals generated using the normal approximation because we have actually calculated Bonferroni simultaneous intervals (so that we have 95% total confidence in the results of the table above). The total confidence in Table 4.5 is less than 67% due to the compounding of the error probability inherent to calculating multiple intervals. If we calculate 95% confidence intervals for the same conditions as Table 4.5, but using the logistic response, then Figure 4.9 shows the result.

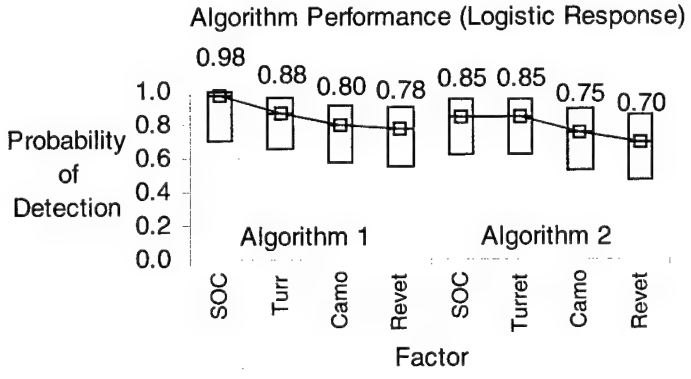

Figure 4.9 Confidence Intervals Using Logistic Regression By Algorithm

To illustrate that these intervals are more representative of the data we are estimating, consider that the sum of the Bernoulli observations has a binomial distribution so the parameter we are estimating (probability of detection) also has a binomial distribution (divided by the number of repeat samples). By generating intervals with the logistic response (with coefficients based upon a Bernoulli distribution) we obtain the correct confidence. The former method is only an approximation of our confidence for the intervals generated. Also, note that the logistic response function intervals are bound between 0 and 1.

4.4.3. Results Using Hypothesis Testing.

Suppose we wish to test the hypothesis that a reduced model is sufficient to explain the variance in the data. Recall that the logistic regression results in only two significant parameters: camouflage and revetments. We can re-estimate our regression coefficients using only these two predictors and evaluate the difference between the deviance statistics of the original (full) model and our new (reduced) model. The deviance for the full model is 123.2 (a measure of our prediction error using this model) and the deviance for the reduced model is 125.9. Since the test statistic, $DEV_{reduced} - DEV_{full} = 2.7$ is less than $\chi^2(.05, p-q) = 11.07$ (where there are p predictors in the full model and q predictors in the reduced model), we accept H_0 and conclude the reduced model is sufficient. To test for goodness of fit, we use the reduced model deviance; since 125.9 is less than $\chi^2(.05, n-q) = 359.5$, we accept H_0 and conclude the model is a good fit. To test whether the effect of the algorithm type is significant, we use the full model and hypothesize:

$$H_0 : \beta_{\text{Algorithm}} = 0$$
$$H_a : \beta_{\text{Algorithm}} \neq 0$$

using the statistic, $z^* = b_{algorithm}/s(b_{algorithm}) = .7$ we see that $|z^*| < z(1-\alpha/2) = 1.96$ so we accept H_0 and conclude that no evidence of a difference between algorithms exists.

4.4.4. Benefits of Improved Methodology.

In summary, the main improvement is the use of logistic regression. We use logistic regression to efficiently identify significant relationships in the data, construct appropriate confidence intervals, and perform hypothesis tests. For our simulated data, the regression technique leads us directly to the most significant results, that the camouflage and revetment factors degrade performance and the algorithms perform nearly the same. Also, we construct intervals that are based on our data distribution and cannot exceed our data boundaries. Finally, we use hypothesis testing to verify that our reduced model (performance based on camouflage and revetments) is sufficient and the two algorithms are not statistically different. We realize these benefits without changing the method of data collection.

4.5. Implementation of Improvements in Phase 2: Data Collection

In this section, we explore the potential benefits of an iterative data collection scheme as well as detailed data characterization. We increase the detail of our characterization of data and demonstrate that the benefits are realized in the analysis phase. We retain the improvements made in the analysis phase, so our updated methodology has improvements in both the collection and analysis phases. Our simulated data is relatively small and does not lend itself well to demonstrating iterative

techniques. We will take an excursion from our example to illustrate the technique, but we will return for the remainder of our research to the simplified problem (three factors with two levels each). When we evaluate our results using a detailed data characterization, we assume that all recommendations from the analysis phase are implemented.

4.5.1. Results Using Iteration.

Recall that the steps we identified in an iterative scheme are screening, characterization, and confirmation. Rather than generate new data to illustrate these steps, consider the following: We return to the example in which we have 7 factors of interest. Our first objective should be to identify any factors that do not affect performance and neglect to vary them for the remainder of testing. To accomplish this we do not need a detailed characterization of every factor, instead we will select two levels for each factor (preferably near its extreme settings) and build a simple fractionated design that will allow us to estimate the single effect of each factor and possibly some (but not all) of the multiple factor effects.

One possible approach would be to generate a 1/8th fractional design with only 16 conditions total. Using these runs we can estimate the single factor effects and the two-factor effects. Suppose three of the factors have negligible effects on performance (including their interaction with other variables and each other), then the results from this experiment might drive us to omit those three factors from further consideration. In the characterization step we could generate a more powerful design for the remaining four factors allowing us to estimate all interactions (a full factorial experiment). This design consists of $2^4 = 16$ conditions plus any additional conditions (like repetitions and center

points); with four center points and two repetitions we have $2 \cdot 16 + 4 = 36$ conditions. Having characterized the variable space, we may suspect that one or two variables have a complex relationship. We might generate a design with two factors but more levels to focus on the nature of their effects. A possible design could be a 5^2 design, or a factorial experiment with two factors and five levels per factor, yielding 25 conditions. Better yet, we could use more advanced designs, like a central composite design which tests five levels in fewer runs (about 11 in total for two factors).

To test every combination of seven factors with five levels would require over 78,000 collected conditions. In the example above we use $16 + 36 + 11 = 63$ conditions to identify factors that do not affect performance, estimate the effects of significant factors, and characterize the nature of non-linear effects. This example can not capture the numerous possible scenarios encountered in ATR performance evaluations, but an experienced analyst can use iteration in this manner to improve the efficiency of testing.

4.5.2. Results Using Detailed Data Characterization.

Recall, our improved results from the analysis phase using simulated data were generated using turret, camouflage, revetments, and an algorithm factor as the four predictor variables in a logistic regression. If we record the azimuth for each image collected, we could include this factor as a predictor in the regression. Experience from past tests reveals that performance tends to degrade as our aspect angle approaches one of the diagonal axes of the target. To include azimuth in such a way that a regression coefficient will make sense, consider this recoding of the azimuth variable: Let the value of the azimuth variable be equal to the absolute value of the smaller angle between the

aspect angle and either the longitudinal or lateral axes of the target. Figure 4.10 illustrates this recoding.

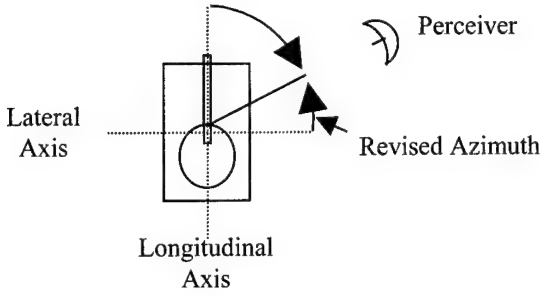

Figure 4.10 Shorter Angle From Target Axes (Revised Azimuth Measure)

We can now regenerate our logistic regression table including both algorithm and azimuth as prediction variables. Table 4.10 shows the results of a logistic regression including all five variables. We see that azimuth does not seem to have much of an effect on the detection probability, even so, we will show that our results improve when we include azimuth in the analysis.

Table 4.10 Logistic Regression Output, Three Factors Plus Algorithm and Azimuth

Logistic Regression Results	All data combined		
	Estimate	Standard Error	p-value (χ^2 test)
Intercept	1.084	0.409	0.008
Turret	-0.256	0.257	0.319
Camouflage	-0.559	0.240	0.020
Revetment	-0.662	0.236	0.005
Algorithm	-0.246	0.151	0.104
Azimuth	**-0.008**	**0.011**	**0.456**

In order to demonstrate improvement, consider again the table containing confidence intervals for each of the collected conditions. If we have included azimuth in the regression, then we can generate a confidence band about the function that represents performance versus azimuth or generate confidence intervals at specific azimuth settings. The graphs in Figure 4.11 show the performance versus azimuth with confidence bands.

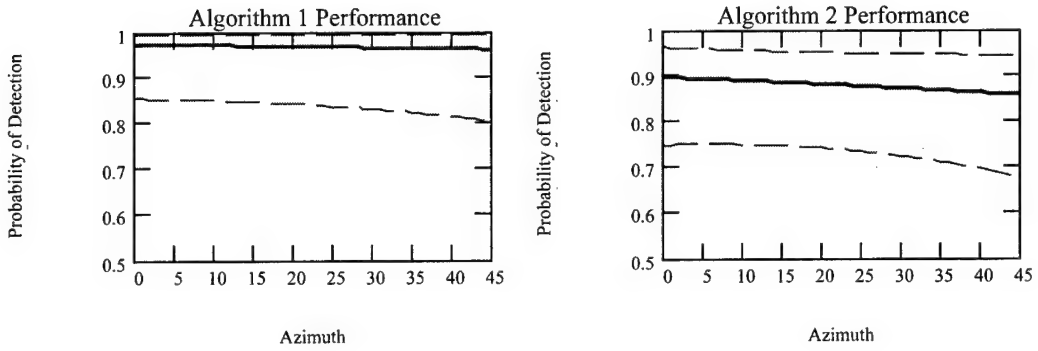

Figure 4.11 Algorithm Performance Versus Azimuth (SOC)

We see in the graph above that Algorithm 1 performs better on average across our four conditions, but we also see that performance degrades slightly in both cases as our recoded azimuth increases. We also have 95% Confidence Bands above and below the mean performance line. In addition to this added information, we are no longer restricted to making inferences that are averaged across azimuth. We can make stronger inferences by estimating performance at any collected azimuth. For instance, Table 4.11 has confidence intervals for mean performance at the highest azimuth setting. From Table 4.11, we can make inferences about performance at each condition for our worst case (azimuth = 45 degrees) and thus establish a lower bound for detection probabilities. We

could also calculate intervals for the best case (azimuth = 0 degrees) or even for some intermediate case.

Table 4.11 Confidence Intervals Using Logistic Response At 45 Degrees Azimuth

Logistic Response C.I.'s	Algorithm 1			Algorithm 2		
	Lower	Mean	Upper	Lower	Mean	Upper
SOC	0.80	0.96	0.99	0.68	0.86	0.95
Turret	0.69	0.87	0.95	0.62	0.81	0.92
Camouflage	0.51	0.71	0.86	0.57	0.77	0.89
Revetments	0.48	0.68	0.83	0.52	0.72	0.86

If we compare these intervals to the intervals in Table 4.5 (azimuth not included), we see now that our former intervals are wider than our new intervals. This is due to the fact that we have explained some of the variance in the data using the azimuth variable and reduced our estimate of noise. If the magnitude of the effect of azimuth on performance is greater, the improvement is more dramatic.

Based on these results, we can postulate that with a reduced estimate of variance, we can detect effects with greater accuracy and make stronger assertions with our hypothesis tests. For example, we could construct a test to determine whether the coefficient associated with the algorithm effect is different from zero (hence, concluding that one algorithm outperforms the other). This test has the form:

$$H_0 : \beta_{\text{Algorithm}} = 0$$
$$H_a : \beta_{\text{Algorithm}} \neq 0$$

using the statistic, $z^* = \beta_{\text{Algorithm}}/s(\beta_{\text{Algorithm}}) = 1.63$ we see that $|z^*| < z(1\text{-}a/2) = 1.96$ so we fail to reject H_0 and continue to conclude that the two algorithms perform identically, however, note the following. In our first hypothesis test using $\beta_{\text{Algorithm}}$, our test statistic is 0.7 (equivalent to a 24.2% probability that our test statistic could occur when β is

actually 0). In our recent trial, our test statistic of 1.63 is associated with only a 5.2% probability that we could observe 1.63 when $\beta = 0$. If we are willing to accept a 12% probability of error (two sided test), we would conclude that $\beta_{\text{Algorithm}} \neq 0$. If the effect of algorithm were greater, we could reject the null hypothesis with our preferred error probability of 5%. The point of this exercise is simply to demonstrate that explaining part of our random variance through background factors can increase our knowledge of the factor-space *without collecting additional data.*

4.5.3. Benefits of Improved Methodology.

The main improvement in this phase is detailed data characterization. We discuss iteration, but do not generate separate data to calculate results using iteration. We find that with a more detailed characterization of data, we have the potential to reduce our estimate of noise, improve the efficiency of confidence intervals (as a result of noise reduction), and we improve our capacity to detect significant effects. In our simulated data, the azimuth effect is relatively small. We believe this scenario is near worst case (no azimuth effect) for a methodology that sets out to collect detailed image information. In a more realistic scenario, there may be multiple, currently unmeasured factors that have large effects on performance. To gain the benefits of these improvements, we expend additional resources to gather detailed information, but we still have not increased or changed the conditions we collect. The most effective means to reduce noise in the test is to collect repetitions of each test condition. Currently, images collected for the same condition, but at different azimuth angles are counted as repetitions. We recommend repeating collection of conditions to obtain repetitions at each azimuth as well. Planning to collect repetitions would occur in the design phase.

4.6. Implementation of Improvements in Phase 1: Test Design

In this section, we demonstrate the benefit of estimating interactions among our variables by including observations from conditions not collected under the one-at-a-time methodology. We use the coefficients from Tables 4.1 and 4.2 to generate random data for the four conditions in our full factorial design not included in the one-at-a-time design (Tables 3.1 and 3.2). First, we compare the results from a full factorial experiment to the results from a one-at-a-time experiment (implementing all techniques from the two previous sections). Then, we compare the results from a fractional factorial experiment with the one-at-a-time experiment so that we are comparing methodologies under equal circumstances (same number of data points). Recall that for our factorial experiment there are eight conditions (see Table 3.2). If we fail to collect data at the conditions that involve two or more factors being varied, our only estimate of performance at these conditions can be constructed using an additive model. To estimate performance at any multiple factor condition, we first calculate the degrade in performance for each of the single effects present in the new condition and add the degrade factors to get a new, estimated degrade factor. For example, the performance at the SOC condition (algorithm 1) is 98% detection, and the performance estimates when turret and camouflage are varied in turn are 88% and 70%, respectively. To estimate the effect of varying turret and camouflage simultaneously, we observe that the degrade factors due to each variable are 10 and 18 percentage points, then we add the factors to get a 28 percentage point degrade. This results in an estimate of 62% probability of detection for this case. Table 4.8 shows the degrade factors and hypothesized performance estimates for all conditions. If the conditions associated with the latter four effects in Table 4.11

are never collected, we have no means to test our hypothesis that the effects are additive. Given the data above, we are in the uncomfortable situation of *estimating* that algorithm 1 outperforms algorithm 2 in the first four conditions and *guessing* that algorithm 2 outperforms algorithm 1 in the latter four conditions.

Table 4.12 Performance Degrade Using an Additive Model for Multiple Effects

Additive Model	Degrade in Performance (Performance estimate %)	
	Algorithm 1	Algorithm 2
SOC	0 (.98)	0 (.85)
Turret	-.10 (.88)	0 (.85)
Camo	-.18 (.80)	-.10 (.75)
Revet	-.20 (.78)	-.15 (.70)
Turret + Camo	**-.28 (.70)**	**-.10 (.75)**
Turret + Revet	**-.30 (.68)**	**-.15 (.70)**
Revet + Camo	**-.38 (.60)**	**-.25 (.60)**
All Three	**-.48 (.50)**	**-.25 (.60)**

At this point, it is not clear which algorithm is superior. Before completing our recommendations, we will consider a design in which we collect data for all conditions. This is our ideal methodology, the final step is to fractionate our full factorial design and complete our improved methodology.

4.6.1. Results Using a Full Factorial Design.

In our simulated data set, we intentionally cause variables to interact to illustrate the potential loss of information inherent to one-at-a-time experimentation. Table 4.12 contains the results from additional observations collected from the multi-factor conditions. It appears from Table 4.13 that algorithm 1 outperforms algorithm 2 in all

but the last condition. If we desire an algorithm that will perform well in most conditions, we might prefer algorithm 1, if we want an algorithm that does not perform worse than 33% detection (three-way interaction), we might prefer algorithm 2.

Table 4.13 Performance Degrade Using Collected Data (Full Factorial Design)

Collected Data	Degrade in Performance (Performance estimate %)	
	Algorithm 1	Algorithm 2
SOC	0 (.98)	0 (.85)
Turret	-.10 (.88)	0 (.85)
Camo	-.18 (.80)	-.10 (.75)
Revet	-.20 (.78)	-.15 (.70)
Turret + Camo	**-.25 (.73)**	**-.37 (.48)**
Turret + Revet	**-.30 (.68)**	**-.47 (.38)**
Revet + Camo	**-.50 (.48)**	**-.42 (.43)**
All Three	**-.85 (.13)**	**-.53 (.33)**

Figures 4.12 and 4.13 contrast the results from both methodologies; confidence intervals are generated using the normal approximation for the one-at-a-time data (additive model) and logistic regression for the factorial data (logistic response model). From the graphs, we see that the additive model is nearly sufficient (with the exception for the three factor effect) for algorithm 1, but grossly overestimates performance for algorithm 2. In both cases, our intervals based on collected data (factorial data) cover the true mean.

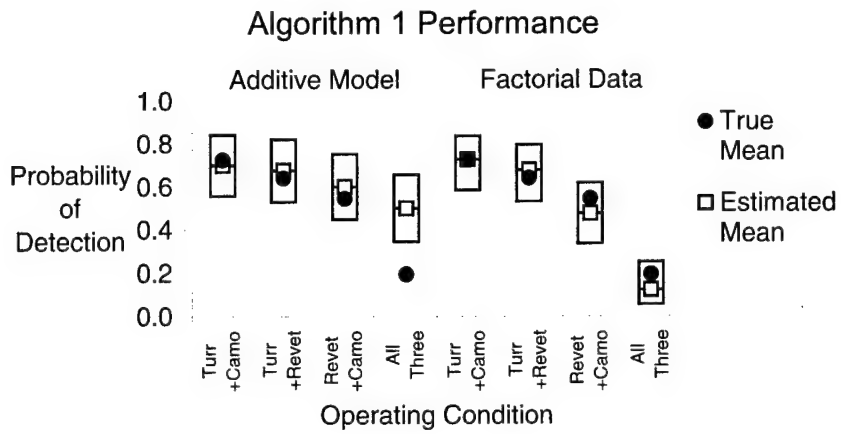

Figure 4.12 Confidence Intervals for Multiple Factor Conditions Using Additive Model and Collected Data (Algorithm 1)

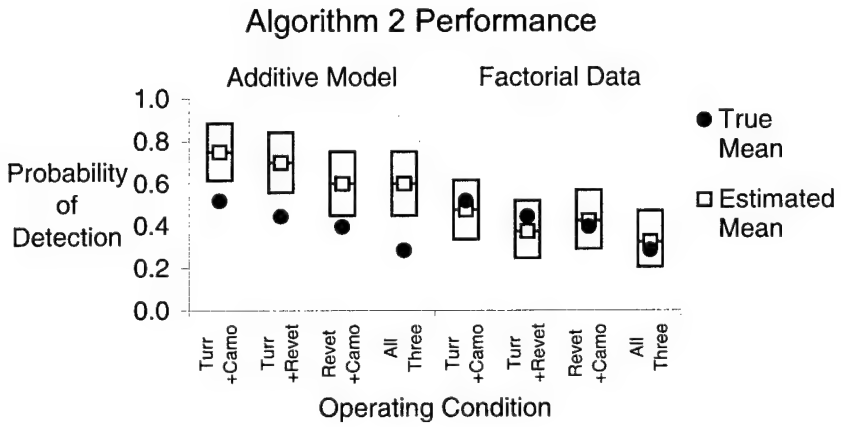

Figure 4.13 Confidence Intervals for Multiple Factor Conditions Using Additive Model and Collected Data (Algorithm 2)

4.6.2. Results Using a Fractional Factorial Design.

In the previous section, we compared factorial data to one-at-a-time data and found that collected estimates are preferable to an additive model. This comparison is biased since we have the benefit of twice as much data for the factorial design. Suppose

we are constrained from collecting all eight conditions. To compare methodologies under similar circumstances, we analyze only a half-fraction of the designed data and compare the analysis with the standard results. If we use only the data from the runs identified in Table 3.3 as a fractional design, we can use the results from a logistic regression to estimate performance for all eight conditions. The graphs in Figures 4.14 and 4.15 show the performance estimates for all eight conditions, compared with the known means. We see in these two graphs that we can successfully capture the true means for all cases without expending more resources, by selecting a more effective and efficient design.

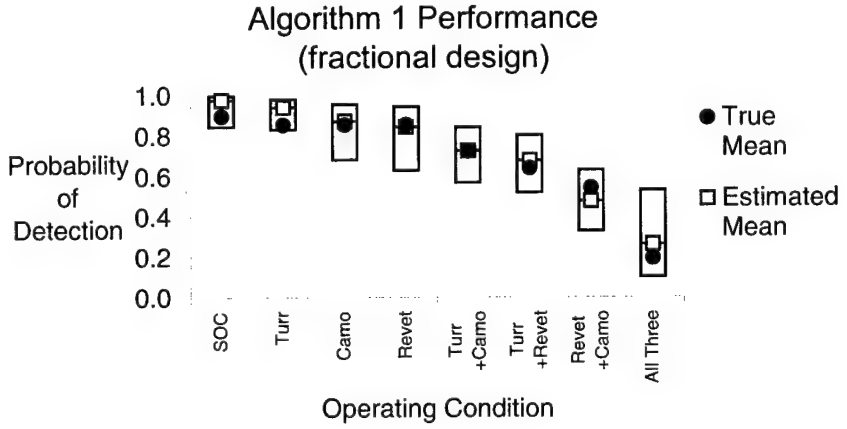

Figure 4.14 Logistic Response Confidence Intervals Using Fractional Design (Algorithm 1)

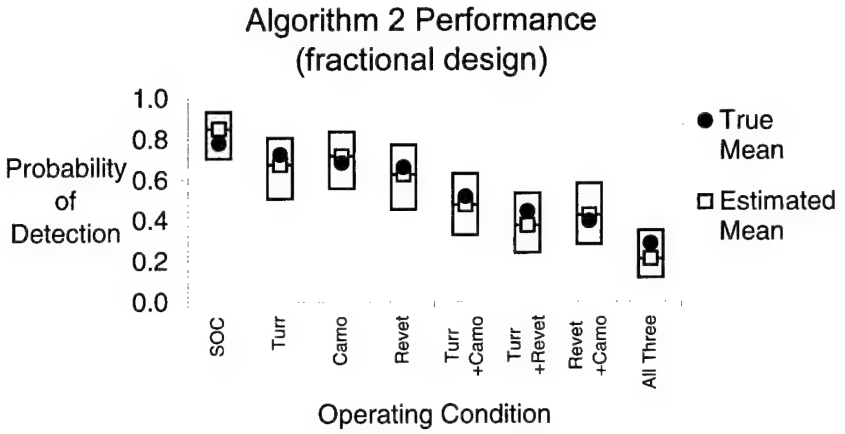

Figure 4.15 Logistic Response Confidence Intervals Using Fractional Design (Algorithm 2)

Again, to be sure the comparisons are fair, we can use the logistic regression technique with the one-at-a-time data to determine whether we cover the true means. The results are in Figures 4.16 and 4.17.

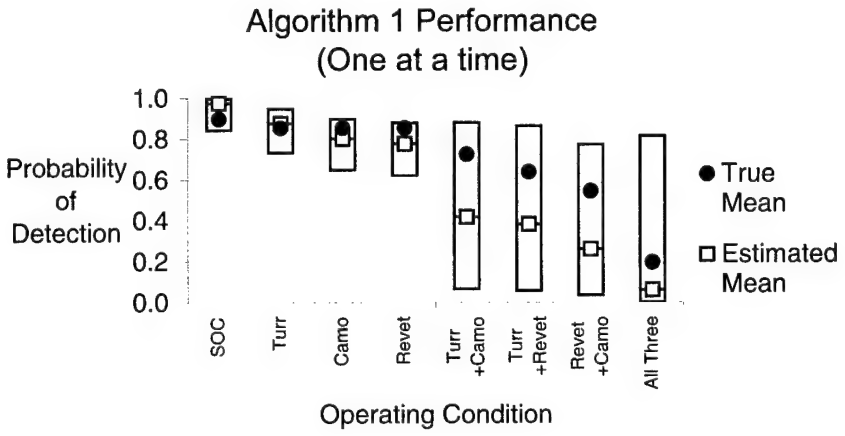

Figure 4.16 Logistic Response Confidence Intervals Using One-at-a-time Design (Algorithm 1)

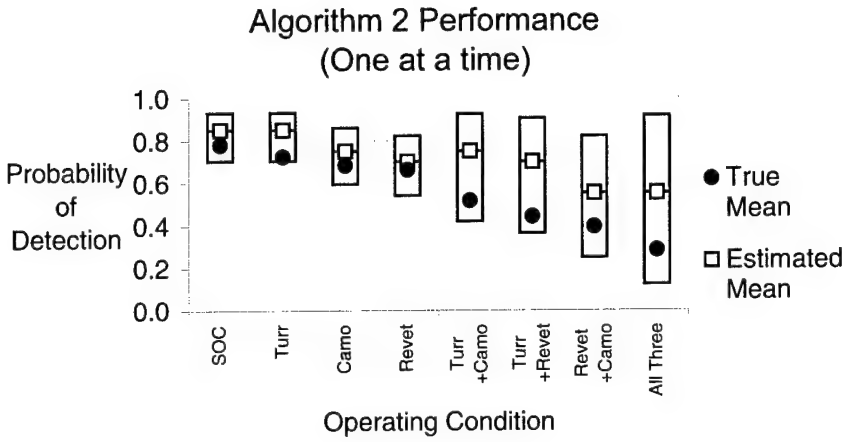

Figure 4.17 Logistic Response Confidence Intervals Using One-at-a-time Design (Algorithm 2)

Using logistic regression, we have improved our intervals to cover the true means (again, demonstrating the superiority of logistic response confidence intervals), but our intervals are still much wider than with our fractional design and our estimates of the means are far from the known values. We can see the improvement in our estimates and inferences (confidence intervals), but the benefits of using designed data are also manifested in hypothesis testing. Recall that the statistic for testing whether the coefficient for the algorithm effect is significant has the form: $z^* = \beta_{Algorithm}/s(\beta_{Algorithm})$, where we compare the statistic to $z(1-\alpha/2) = 1.96$. With the one-at-a-time data, our statistic is approximately 0.7, leaving us inconclusive as to whether one algorithm is better than another (across all conditions). The statistic using the fractional data is 3.26 (which is much greater than our critical value of 1.96) and we can now conclude, with 95% confidence, that algorithm 1 outperforms algorithm 2 on average. Using the logistic regression coefficient for the algorithm effect (-0.56), we estimate that the resultant

degrade in the odds of detection for any condition due to selecting algorithm 2 instead of algorithm 1 is approximately 57% (exp(-.56) = .57). This means that if algorithm 1 detects with 90% accuracy (odds = 9 in 10 chance), we estimate that algorithm 2 would detect with about 75% accuracy for the same condition (odds = $9 \cdot (.57)^2 = 2.9$ in 3.9 chance). To verify this, we set the azimuth to 5 degrees, and find that our estimate using the regression coefficients for algorithm 1 performance is 90% (for the camouflage only condition). Our estimate for algorithm 2 is 76%, very close to 75%, as we estimated from Figure 4.11. The tests we can perform and inferences we can make about the true performance are flexible and numerous; reporting results in this manner is statistically more rigorous than making assertions based on graphs and tables.

4.6.3. Benefits of Improved Methodology.

We conclude the implementation of our recommendations by modifying our data design in two ways. First, we consider a factorial design in which we select levels for each factor and construct conditions that cover the entire spectrum of possible factor combinations. Next, we fractionate our design to reduce the number of conditions necessary to gather the information we need. Finally, we show that the intervals from our fractional factorial design cover our known parameters whereas an additive model of performance may not. By using our methodologies with simulated data, we demonstrate that the potential for improvement does exist. This potential is manifested in the benefits realized for our simulated data, which only represents one possible set of outcomes for our test scenario. It is possible that the benefits are not evident with another set of outcomes. We show in chapter 5 that the benefits of the improved methodology persist under a variety of outcomes for our ATR evaluation scenario.

5. SENSITIVITY OF BENEFITS TO VARIANCE IN PERFORMANCE DATA.

In this chapter, we vary the parameters of our truth model and simulate a wide range of possible outcomes for our test scenario. We implement our current and improved methodologies and estimate the change in the benefits of our recommendations due to variance in the truth model. The chapter is organized as follows.

- Variation in performance data
- Simulation of variation
- Characterization of a methodology
- Measurement of the benefits of a methodology
- Sensitivity analysis
- Results summary

Our objective is to demonstrate that for a variety of data, the benefits of our improved methodology persist.

5.1. Variation in Performance Data

We have established that for one set of data, the benefits of our improved methodology exist. For our test scenario (3 factors, 2 levels per factor plus a background factor), there are endless possible outcomes in the data. Recall that our simulated data was built by generating observations from a Bernoulli random variable, with parameters determined by our known coefficients (truth model). Suppose our known parameters can vary. If we change the values that are the coefficients for our truth model, the performance for each condition changes and we generate random observations from a

different set of distributions. Furthermore, we can vary the coefficient for the azimuth factor and vary the magnitude of the azimuth effect. We can also vary the difference between algorithms, even simulating a range of possible differences between algorithm performance levels. Recall that in chapter 4 we simulate data using two distinct truth models (one for each algorithm) which allows us to have different detection probabilities for each algorithm. In this chapter we use one model with an algorithm factor.

5.2. Simulating Variation in Performance Data

Here we discuss the method by which we generate multiple data sets that span the possible outcomes of our test scenario. In general, we take the following steps.

- Identify key coefficients to vary and construct a prototype truth model
- Select levels for each coefficient and build a full factorial design using coefficients as factors
- Fractionate the design and generate random observations from each unique truth model (sensitivity design points)

In order to encompass the broadest set of possibilities within our resources, we utilize an experimental design approach.

5.2.1. Key Coefficients for Variation.

When we vary the value of a coefficient in a logistic response function (our prototype truth model), we are not varying the level of the factor (e.g., changing the coefficient for turret does not mean the level of articulation changes from zero to 10 degrees). Rather, it changes the effect on performance due to the factor associated with the coefficient. In other words, we can force the degrade in performance due to our

factors (turret, camouflage, revetments, and azimuth) to increase or decrease. If the coefficient for a factor is near zero, changing the level of the factor does not affect performance. As the coefficient decreases from zero, the factor degrades performance. The entities that we vary in our sensitivity analysis are the coefficients for our original factors (turret, camouflage, revetments, and azimuth). Also, we include coefficients for the mean response (the intercept term in a standard regression), two and three-factor interactions, an algorithm effect, and interactions between the algorithm factor and two-factor interactions. Based on these entities, we have identified 13 coefficients that we can vary in our truth model.

5.2.2. Coefficient Variation Levels.

We only select levels for our coefficients that induce a degrade in the detection probability. We select levels such that a given factor or interaction will effect either a very small degrade in performance or a very large degrade. Table 5.1 shows our coefficients and levels.

Table 5.1 Logistic Response Function Coefficients Varied in Sensitivity Analysis

Coefficient Varied	Low level	High level
Mean Response	0 (50% detection)	4 (98% detection)
Turret	-0.1 (20% approximate reduction in detection odds)	-1.1 (90% approximate reduction in detection odds)
Camouflage		
Revetment		
Two-way Interactions		
Three-way Interaction		
Algorithm		
Algorithm/Turret/Camo		
Algorithm/Turret/Revetment		
Algorithm/Camo/Revetment		
Azimuth	-0.009 (1% approximate reduction in detection odds per degree)	-0.120 (10% approximate reduction in detection odds per degree)

Varying the mean response coefficient changes the average performance across all conditions, the location of the mean probability. Varying single factor coefficients changes the effect of those factors. Changing interaction coefficients induces complex relationships between the factors involved in that interaction. Using these 13 coefficients and levels, there are $2^{13} = 8192$ unique combinations. Each combination is a set of coefficient values that can be input to the logistic response function to form a unique truth model. We use fractionation to reduce this number to a manageable size.

5.2.3. Truth Model Set.

We select a 1/256 fractional design for sensitivity analysis. The result is 32 coefficient sets that represent a broad cross section of the numerous possibilities (resolution IV). For each design point (a set of coefficients) we use our prototype truth model to generate 32 separate data sets of random observations. Each data set consists of 40 observations per original test condition (Figure 4.3). With 8 test conditions and 40 observations per condition (and two algorithms), we have $8 \cdot 40 \cdot 2 = 640$ observations for each truth model. Figure 5.1 illustrates the data set generation process. Once we generate these data sets, we can construct an experiment to compare methodologies.

5.3. Characterization of Methodologies

In order to demonstrate the gradual improvement in results that come from the stepwise implementation of our recommendations, we select five methodologies total for comparison (see Figure 4.1).

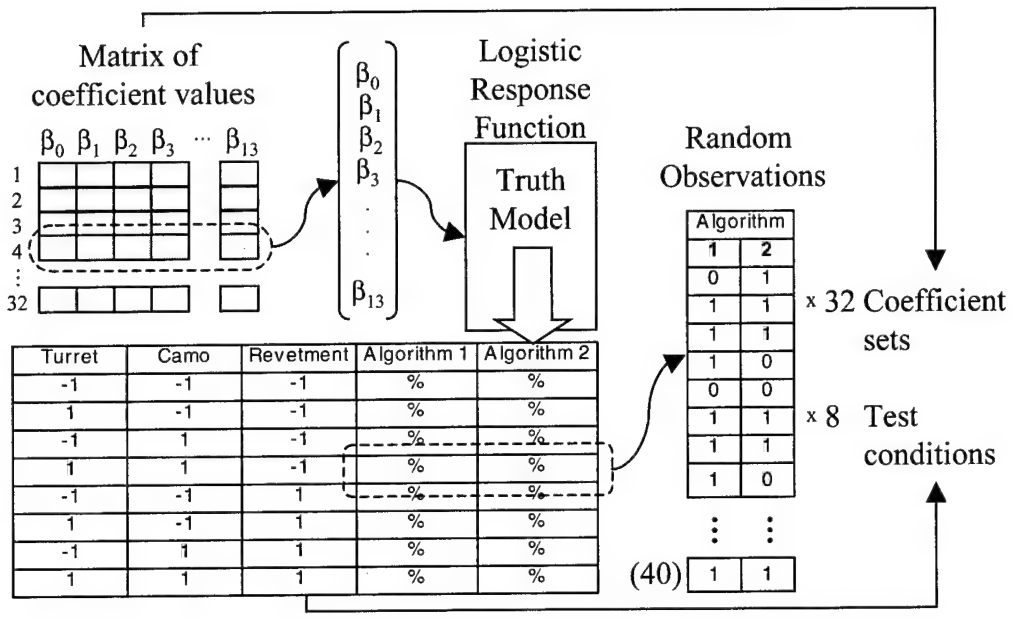

Figure 5.1 **Process for Generating Multiple Random Data Sets from Varying Truth Model**

- Method 1: A standard methodology that does not include a formal statistical analysis of data, does not include background variance (azimuth) as a prediction variable, and *only utilizes data from the conditions collected under a one-at-a-time experiment*. Estimates are based on averaging and the use of an additive model for conditions not collected. Intervals are generated using the normal approximation to the binomial distribution.
- Method 2: An improvement on the above method that is identical except logistic regression is used to analyze data, make estimations, and generate confidence intervals.
- Method 3: A further improvement that includes azimuth as an additional prediction variable.

- Method 4: The revised methodology from chapter 4 that adopts an experimental design approach to selecting conditions for collection. The data used for this methodology are from the *half-fraction experiment* in Table 3.3.
- Method 5: The revised methodology using a full factorial design for collection *(data from all eight conditions are used)*. This method is the ideal approach with theses improvements, without regard to resources.

Each of these methodologies uses a portion or all of the simulated data sets to calculate performance results. Rather than perform the detailed analysis from chapter 4, we identify a few simple calculations to measure the benefit of using a methodology.

5.4. Measuring the Benefits of a Methodology

In order to compare methodologies, we need a means to measure the merit of a methodology. First, we define the quality of results: The quality of the results of an evaluation is a consequence of our success in minimizing, accurately estimating, and clearly communicating uncertainty. We provide justification for addressing uncertainty in this manner and introduce three calculations that measure the quality of our results, as we have defined it above.

5.4.1. Addressing Uncertainty.

When we use a model to explain the variance in a test, some of the variance is due to the test factors we vary, and the rest we attribute to random error. We minimize this random error by including background factors in the analysis, using orthogonal data designs, and collecting more data. The result is increased confidence in our results. The

easiest means of illustrating this is with confidence intervals. If we use some method to decrease our error estimate, the result is smaller confidence intervals.

Estimates of uncertainty are based on the assumptions of a statistical model or statistical test. Violating an assumption degrades the accuracy of our error estimates. Gross violations render our estimates meaningless. We ensure accurate estimates of uncertainty by checking our assumptions and accounting for gross violations.

Results are unclear if the communication of uncertainty does not add to our understanding of the nature of the data (or actually detracts from our understanding). Inaccurate estimates of uncertainty or large overall uncertainty lead to unclear results. By ensuring the two former issues are resolved, we are not hindered in clearly communicating uncertainty.

5.4.2. Measure 1: Estimation Error.

For convenience, we choose measures that are readily available to us but still address the objectives above. The first measure is built upon the distance from the true (known) performance parameter and our estimate based upon collected data (Figure 5.2). In each methodology we build a model of performance and estimate detection for each of the eight conditions. The average distance from the true parameter across all conditions will be our first performance estimate. Given a table of numbers that estimate the average detection probabilities for each of the eight conditions (and for both algorithms), like Table 4.8, and given a matching set of known parameters for each condition, our measure (average error) is calculated with the following equation:

$$Average\ Error = \sum_{i=1}^{8} |\pi_i - p_i| \qquad \textbf{(5.1)}$$

where π_i is the known detection probability for condition i and p_i is the estimated detection probability for test condition i.

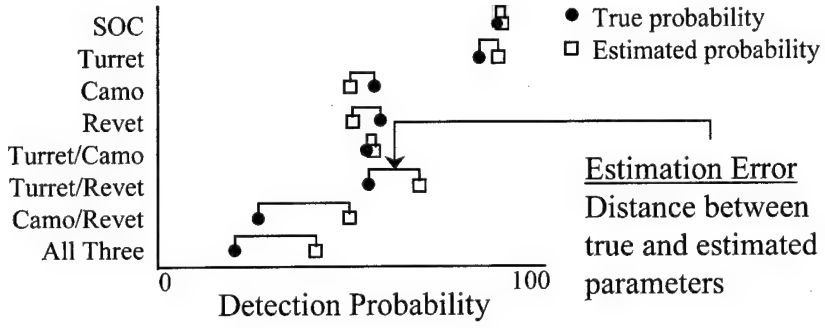

Figure 5.2 Illustration of Estimation Error Measure

5.4.3. Measure 2: Parameter Coverage.

The second measure is based upon whether or not confidence intervals constructed according to the techniques in a given methodology successfully cover the known detection probabilities. Clearly, we believe that there will be some correlation between this measure and the first measure since methodologies that result in smaller estimation error should also result in a higher likelihood of covering the true mean in a confidence interval. We justify this second measure by pointing out that two methodologies that result in similar estimates of the mean response can be differentiated by testing whether one or both failed to cover the true mean. Furthermore, we point out that it is even possible for one method to result a smaller estimation error than another, but actually fail to cover the true mean in an interval while the other method succeeds.

The measure is calculated by summing the number of successes in covering the true response across the eight conditions (Figure 5.3), as shown by the following equation:

$$Coverage = \sum_{i=1}^{8} C_i \qquad \textbf{(5.2)}$$

where,

$$C_i = \begin{cases} 1 & \text{if } \pi_i \text{ is covered by confidence intervals} \\ 0 & \text{otherwise} \end{cases}$$

This measure is bounded by 0 and 8.

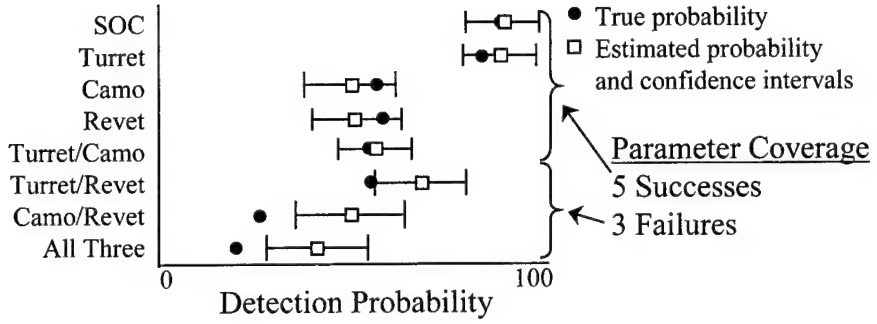

Figure 5.3 Illustration of Parameter Coverage Measure

5.4.4. Measure 3: Interval Efficiency.

We recognize a further need to analyze confidence intervals since one objective is to minimize uncertainty. Two methodologies may have similar estimation error and both capture the true parameter, but it is more desirable that a method generates intervals only large enough to capture the target performance probability. We therefore desire smaller confidence intervals (this is achieved by reducing our estimate of noise). Rather than measure only the width of the interval, we calculate the average number of true parameters we cover per 10 percentage points of interval width (Figure 5.4). This

measure has the effect of rewarding methodologies that capture our true parameters with small intervals but penalizing methods that miss the true parameters or generate unnecessarily large intervals. The equation below shows the method for calculating this measure:

$$Efficiency = \frac{\frac{1}{8} \cdot \sum_{i=1}^{8} i}{\frac{1}{8} \cdot \sum_{i=1}^{8} (u_i - l_i) \cdot 10} \qquad (5.3)$$

where i = 0 or 1 depending on whether the confidence interval associated with the i^{th} condition captures the true parameter, and u_i and l_i are the values of the upper and lower confidence interval limits respectively.

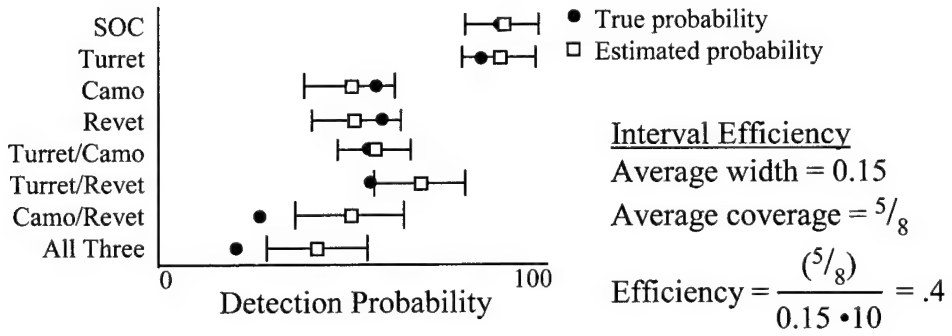

Figure 5.4 Illustration of Interval Efficiency Measure

With these measures, we can quickly analyze the difference between the results from separate methodologies, without accomplishing the burdensome calculations necessary to perform the tests in chapter 4. There are many other measures we could have constructed, but these measures address our basic objectives in choosing a methodology: minimization and accurate characterization of uncertainty. Other calculations that

measure our achievement of these objectives will necessarily be correlated with our three chosen measures.

5.5. Results of Sensitivity Analysis

Using the measures identified in the previous section, we can use standard regression to test the effect of all of our potential prediction variables (coefficients and components). Specifically, we can test what the average effect is of varying the magnitude of coefficients and see how different test results affect our ability to estimate results, without regard to methodology. Also, we can test how the different components of our methodologies (regression, background variance estimation, and experimental design, both fractional and full factorial) affect the quality of our results. Finally, we can analyze how the different test results (due to magnitude of factor effects) affect the difference between methodologies.

- Sensitivity analysis approach
- Impact of varying coefficients
- Impact of varying methodology components

Using the results from these steps, we can identify the most significant relationships in our sensitivity dataset.

5.5.1. Method of Analysis.

We use standard normal regression to analyze the data generated by our three measures (since the measures are not binary). The prediction variables for our regression are the *methodology components* and the *coefficient levels*. To include the methodology components, we use variables coded [0,1] for each component, as shown in Table 5.2.

Table 5.2 Coding Scheme for Methodology Component Variables

Methodology	Methodology Components			
	Logistic Regression	Background Variance	Factorial Design	Full Factorial Design
Current	0	0	0	0
Intermediate 1	1	0	0	0
Intermediate 2	1	1	0	0
Improved	1	1	1	0
Ideal	1	1	1	1

The coefficient variables are coded [-1,1] for the low and high levels. Since we must implement each of our five methodologies on all 32 data sets, there are $5 \cdot 32 = 160$ conditions for which we generate data with our new measures. We perform a standard regression with the 4 component variables and 13 coefficient variables using our 3 measures as dependent variables.

5.5.2. Impact of Varying Coefficients.

Before comparing methodologies, it may be useful to understand which of the coefficients cause variance in our measures across all methodologies. Using standard regression with our measures as responses and coefficients as predictors we generate the results in Table 5.3. We see the estimation error grows as we increase the magnitude of several of our effects. As the mean response moves away from the center (0.50), we estimate the mean detection probability with less accuracy. Increasing the degrade due to two-factor effects also increases estimation error for one of the three elements. Increasing the degrade due to the three factor interaction degrades estimation accuracy, and so does increasing the degrade due to azimuth.

Table 5.3 Coefficient Effects, all Methodologies, Response: Estimation Error

Regression Results	Coefficient	Standard Error	t* statistic	Probability t > |t*|
Intercept	0.052	0.011	4.620	<.0001
Mean response	0.056	0.006	9.390	**<.0001**
Turret effect	0.001	0.006	0.090	0.929
Camo effect	-0.005	0.006	-0.840	0.399
Tur/Camo Effect	0.017	0.006	2.770	**0.006**
Revetment effect	0.004	0.006	0.650	0.516
Turr/Revet effect	0.002	0.006	0.360	0.718
Camo/Revet effect	0.004	0.006	0.600	0.551
Tur/Cam/Rev effect	0.012	0.006	1.980	**0.048**
Algorithm shift	0.002	0.006	0.410	0.684
Tur/Cam/Alg effect	-0.010	0.006	-1.610	0.108
Tur/Rev/Alg effect	-0.010	0.006	-1.670	0.096
Cam/Rev/Alg effect	0.016	0.006	2.730	**0.007**
Azimuth effect	-0.031	0.006	-5.150	**<.0001**

Legend: - t* is the students' t-test statistic
- Probability t > |t*| is the probability of obtaining a statistic greater than t from a t-distribution

Table 5.4 shows that no effects seem to affect our ability to cover the known parameter with our intervals. In Table 5.5, we see that the only elements that affect the interval efficiency are the location of the mean response, and the difference in one of the two-factor effects across algorithms. We seem to have better success when our mean response is centered. The negative coefficient for the mean response effect magnitude means the intervals are less efficient as they move away from the center. This makes sense because our error estimate is largest at 0.50 and our intervals are wider, thus increasing our chance of covering the parameter.

Table 5.4 Coefficient Effects, all Methodologies, Response: Parameter Coverage

Regression Results	Coefficient	Standard Error	t^* statistic	Probability $t > \|t^*\|$
Intercept	7.225	0.325	22.210	<.0001
Mean response	-0.213	0.174	-1.220	0.223
Turret effect	-0.075	0.174	-0.430	0.667
Camo effect	-0.025	0.174	-0.140	0.886
Tur/Camo Effect	-0.288	0.174	-1.650	0.099
Revetment effect	0.013	0.174	0.070	0.943
Turr/Revet effect	-0.238	0.174	-1.370	0.173
Camo/Revet effect	-0.138	0.174	-0.790	0.430
Tur/Cam/Rev effect	-0.250	0.174	-1.440	0.152
Algorithm shift	0.175	0.174	1.010	0.315
Tur/Cam/Alg effect	0.038	0.174	0.220	0.829
Tur/Rev/Alg effect	-0.188	0.174	-1.080	0.282
Cam/Rev/Alg effect	-0.163	0.174	-0.930	0.351
Azimuth effect	0.175	0.174	1.010	0.315

Legend: - t^* is the students' t-test statistic
- Probability $t > |t^*|$ is the probability of obtaining a statistic greater than t from a t-distribution

Table 5.5 Coefficient Effects, all Methodologies, Response: Interval Efficiency

Regression Results	Coefficient	Standard Error	t^* statistic	Probability $t > \|t^*\|$
Intercept	0.387	0.034	11.340	<.0001
Mean response	-0.070	0.018	-3.820	**0.000**
Turret effect	0.006	0.018	0.330	0.745
Camo effect	-0.019	0.018	-1.060	0.288
Tur/Camo Effect	-0.018	0.018	-0.990	0.323
Revetment effect	-0.028	0.018	-1.520	0.131
Turr/Revet effect	-0.012	0.018	-0.660	0.509
Camo/Revet effect	-0.012	0.018	-0.650	0.518
Tur/Cam/Rev effect	-0.010	0.018	-0.540	0.591
Algorithm shift	-0.016	0.018	-0.860	0.391
Tur/Cam/Alg effect	-0.008	0.018	-0.420	0.674
Tur/Rev/Alg effect	0.037	0.018	2.000	**0.046**
Cam/Rev/Alg effect	-0.003	0.018	-0.190	0.851
Azimuth effect	0.007	0.018	0.380	0.704

Legend: - t^* is the students' t-test statistic
- Probability $t > |t^*|$ is the probability of obtaining a statistic greater than t from a t-distribution

5.5.3. Impact of Varying Methodology Components.

Another useful step in our analysis is to view the average effect of our methodology components without regard to where the effects manifest themselves among the coefficient effects. For instance, we can view the effect of using logistic regression across all sensitivity data sets, though we do not view whether the benefits are linked to a particular coefficient effect (like the magnitude of the three-factor interaction). Table 5.6 shows the regression results using the estimation error.

Table 5.6 Component Effects, all Coefficients, Response: Estimation Error

Regression Results	Coefficient	Standard Error	t* statistic	Probability t* > |t|
Intercept	0.098	0.008	11.930	<.0001
Full factorial	-0.045	0.011	-4.250	**<.0001**
Designed data	-0.006	0.011	-0.530	0.597
Background variance	-0.001	0.011	-0.130	0.897
Logistic Regression	-0.013	0.011	-1.190	0.236

Legend: - t* is the students' t-test statistic
- Probability t > |t*| is the probability of obtaining a statistic greater than t from a t-distribution

In Table 5.6, we see the average distance from the true parameter is about 10 percentage points on average. The coefficient for the full factorial component (associated solely with the ideal methodology) tells us we increase our estimation error if we do not use a full factorial designed experiment but we halve our error if we use a full factorial design. The other components do not have a strong effect across all sensitivity data sets, but we

may see effects become manifest when we consider underlying conditions (coefficient effects). Table 5.7 shows regression results for our second measure.

Table 5.7 Component Effects, all Coefficients, Response: Parameter Coverage

Regression Results	Coefficient	Standard Error	t^* statistic	Probability $t^* > \|t\|$
Intercept	5.156	0.170	30.300	<.0001
Full factorial	0.469	0.220	2.130	0.034
Designed data	0.063	0.220	0.280	0.776
Background variance	0.031	0.220	0.140	0.887
Logistic Regression	2.031	0.220	9.250	**<.0001**

Legend: - t* is the students' t-test statistic
- Probability t > |t*| is the probability of obtaining a statistic greater than t from a t-distribution

Table 5.7 reveals that the average number of true parameters successfully covered with our intervals is approximately 5, with 8 possible. The effect coefficients tell us that we gain two parameters for a total of 7 out of 8 (on average) if we use the logistic regression technique. Again, there may be effects due to the other components that are not observable in this table. Table 5.8 shows results for our last measure. In Table 5.8, we see the mean interval efficiency is about 0.40, or approximately 5 parameters captured out of 8 and an average interval width of about 15 percentage points (see Figure 5.4). The coefficients above reveal that a full factorial design improves the efficiency of our intervals, but the logistic regression technique is penalized for inflating our interval width in the process of covering more true parameters (see Table 5.7). These results only reveal the average effects of methodology components across all sensitivity data sets.

Table 5.8 Component Effects, all Coefficients, Response: Interval Efficiency

Regression Results	Coefficient	Standard Error	t^* statistic	Probability $t^* > \|t\|$
Intercept	0.392	0.020	19.350	<.0001
Full factorial	0.188	0.026	7.170	**<.0001**
Designed data	-0.022	0.026	-0.840	0.401
Background variance	0.005	0.026	0.190	0.853
Logistic Regression	-0.136	0.026	-5.210	**<.0001**

Legend: - t* is the students' t-test statistic
- Probability t > |t*| is the probability of obtaining a statistic greater than t from a t-distribution

5.6. Sensitivity Analysis Results Summary

In the final stage of sensitivity analysis, we include both the coefficient level magnitudes and methodology components as predictors in a standard regression. Analysis similar to that performed in the previous section can be used to generate the result summary in Table 5.9.

Table 5.9 Summary of results from sensitivity regression analysis

Measure	Significant Main Effects	Significant Interaction Effects
Estimation Error	Full factorial design component	- Designed Experiment component & mean response coefficient - Logistic regression component & Turret/Camouflage coefficient
Parameter Coverage	Full factorial design component Logistic regression component	
Interval Efficiency	Full factorial design component	- Designed Experiment component & mean response coefficient - Logistic regression component & Turret/Camouflage coefficient

Table 5.9 clarifies that the variance in the estimation error, for instance, can be explained by three variables: whether or not a full factorial design is implemented, whether a designed experiment is implemented (depending on the location of the mean response), and whether logistic regression is used (depending on the magnitude of a two factor effect). The utility of generating results in this fashion is that we can now graph these relationships and remain confident that we only display the most significant portion of the variance in our methodology quality measures. Now we address the summary results for each measure, using graphs to illustrate the relationships between our predictors (coefficients and components) and our responses (estimation error, parameter coverage, and interval efficiency). Even this sensitivity analysis illustrates how regression allows us to quickly narrow our attention to the significant test phenomena.

5.6.1. Estimation Error Results.

The graph in Figure 5.5 illustrates the relationships identified in Table 5.9 for our first measure. In all following box-plots, each data point represents an observation (using one of our three measures) from one of the 32 data sets, after applying a methodology. The box represents the inner-quartile range, or, the 25^{th} and 75^{th} percentiles. The upper and lower lines are the 10^{th} and 90^{th} percentiles. Figure 5.5 shows that the estimation error decreases significantly when a full factorial design is implemented (ideal methodology). We have included lines in the graph that show the trend in the data for the cases when the three-factor effect (turret, camouflage, and revetments) has a small and large magnitude. The effect of the three-factor effect element is small here, but we want to illustrate that this element does slightly shift our mean error upward for the cases where a full factorial design is not used.

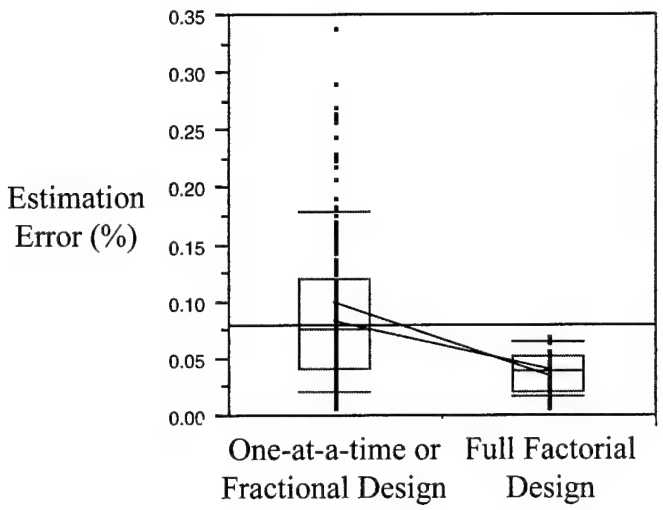

Figure 5.5 Effect of Full Factorial Design on Estimation Error

The box-plot in Figure 5.6 shows that the average effect of using a designed experiment (improved and ideal methodologies) decreases the estimation error. This statement is insufficient, however, to describe the effect of this component because it interacts with the mean response location coefficient. In other words, to provide an accurate estimate of the effect of designed experimentation, we have to know the location of the mean response. Figure 5.6 shows that the error increases when the location of the mean response is far from center and a designed experiment is not used (current and intermediate methdologies). This may be due to the fact that our response is bounded by 0 and 1, thus bounding a degrade between 0 and 0.50 when the response is centered but only bounding it between 0 and 0.98 when the response is far from center.

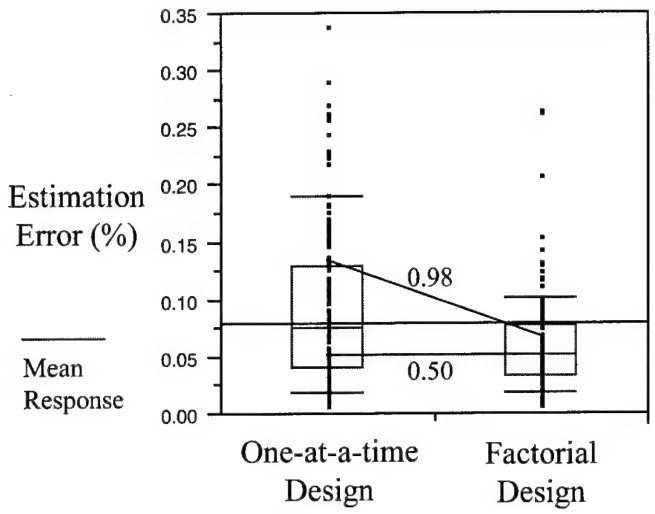

Figure 5.6 Effect of Factorial Design on Estimation Error

The result is that when we have interactions among factors, the magnitude of the degrade can be larger for the non-centered response and result in larger errors when an additive model is used. Figure 5.7 demonstrates the effect of the logistic regression component. The average effect of using logistic regression (which is used in all but the current methodology) is also a reduction in estimation error. Again, there is an interaction that results in higher estimation error when a two-factor effect is large and regression is not used (current methodology). This can be explained by pointing out that the logistic regression approach is not based in an additive model but actually uses the binomial distribution to postulate the effect of two factor interactions. This extra knowledge about our response can actually reduce our estimate of error.

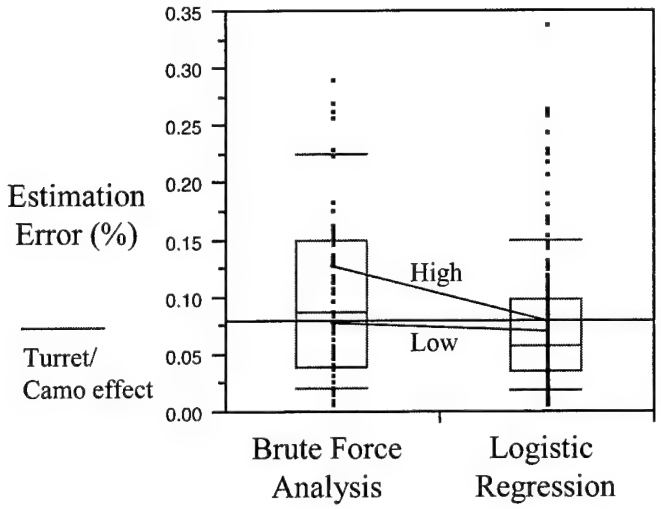

Figure 5.7 Effect of Logistic Regression on Estimation Error

5.6.2. Parameter Coverage Results.

The effect of methodology on the number of true parameters covered with confidence intervals can be explained easily. The average number of parameters covered increases for the ideal methodology where a full factorial design is collected, and decreases for the current methodology where a normal approximation is used to generate intervals. All other methods have roughly the same performance. Figure 5.8 shows the results for the parameter coverage measure in two graphs. The first graph (left) separates the coverage data by whether a full factorial design is used or not. The first box-plot includes data from all but the ideal method, which are in the second box-plot. The second graph (right) separates data by whether logistic regression is used. The first box-plot is formed using only data from the current methodology, the second box-plot contains data from the remaining methodologies.

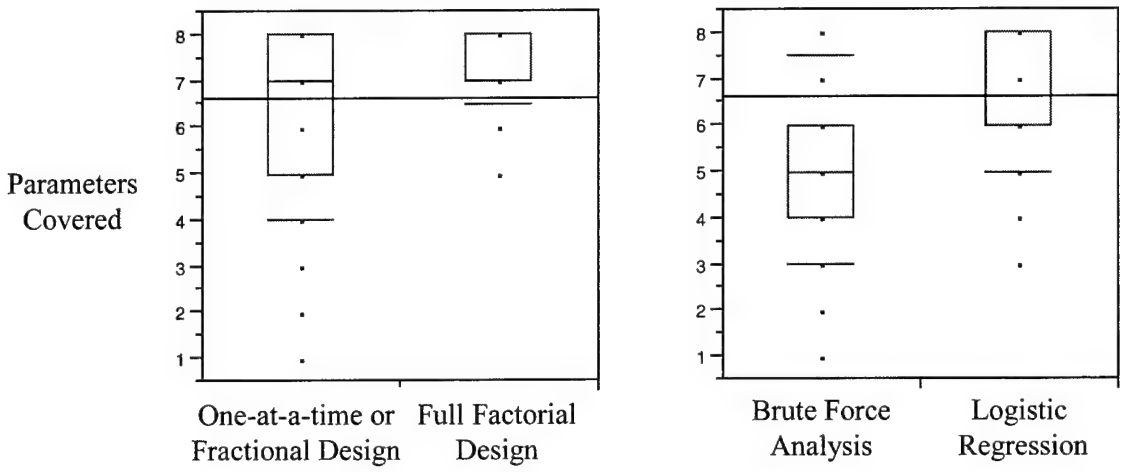

Figure 5.8 Effects of Full Factorial Design and Logistic Regression on Parameter Coverage

5.6.3. Interval Efficiency Results.

Figure 5.9 shows the average effect of using a full design increases the ratio of parameters covered to interval width, essentially rendering our confidence intervals more efficient. The lines represent the turret/camouflage effect and are there to demonstrate that the two-factor effect coefficient slightly affects the magnitude of the full design component effect. Figure 5.10 shows the average effect of using designed experiments is zero, but there is an interaction with the location of the mean response. It seems that when the mean response is centered (low), the use of designed experiments produces slightly less efficient intervals, and when the mean response is far from center, it produces more efficient intervals. Figure 5.9 is arranged similar to the first graph in Figure 5.8. Figure 5.10 separates data by whether some form of factorial design is used. The box-plot on the left contains data from the first three methodologies and the others are in the box-plot on the right.

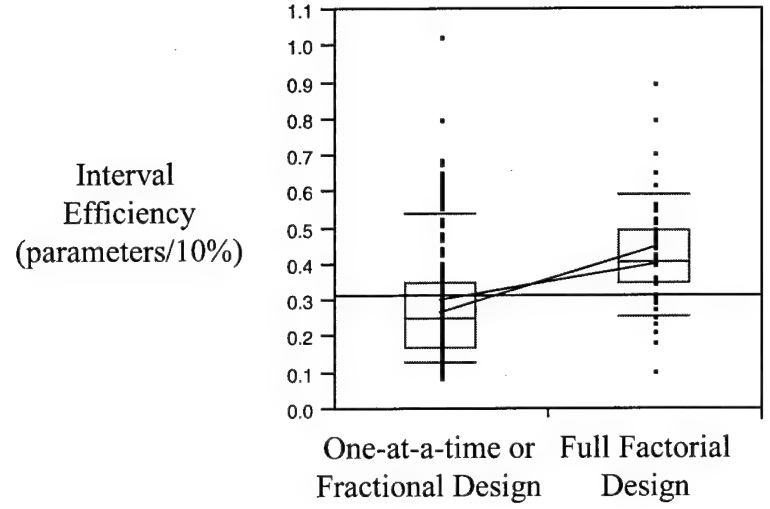

Figure 5.9 Effect of Full Factorial Design on Interval Efficiency

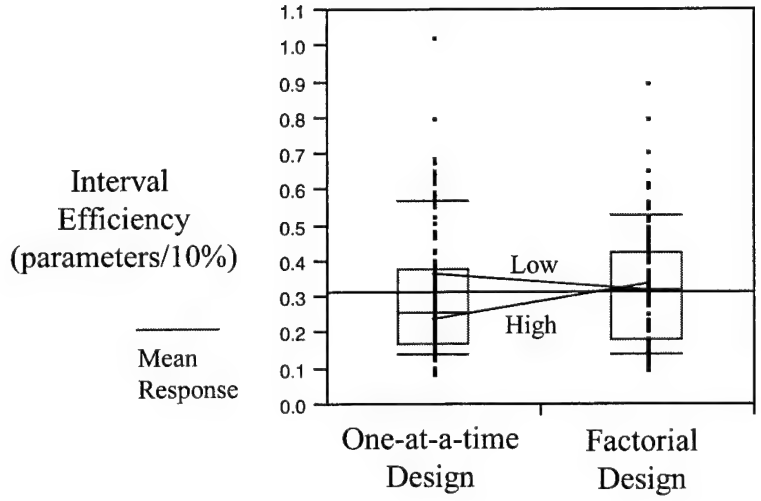

Figure 5.10 Effect of Designed Experiments on Interval Efficiency

In Figure 5.8, there is no effect on parameters captured due to the factorial design component, so we may assume that we capture roughly the same number of parameters with or without a factorial design. This means that the interval widths are larger near

center and smaller off center with the designed experiments component. This may actually make sense because the factorial design methods (improved and ideal) capture the effects of interactions and increase the confidence widths appropriately while the other methods do not. However, when the mean response is far from center, this effect is nullified by the upper boundary (1) and there is a slight improvement over the intermediate methods.

Figure 5.11 shows that the average effect of logistic regression degrades the efficiency of our intervals. We see that when there is a large two-factor effect present, the regression component has no effect. When there is no interaction between main effects, however, the regression intervals are less efficient. This can be explained by considering that the regression intervals cover more true parameters by increasing the interval width which is good when interactions are present, but are less efficient when no interaction is present.

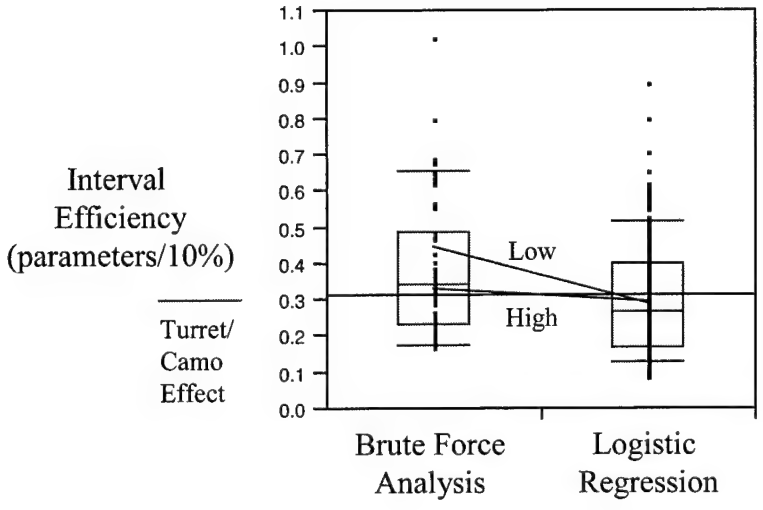

Figure 5.11 Effect of Logistic Regression on Interval Efficiency

5.6.4. Utility of Experimental Design.

Having identified and investigated the effects of individual components of our set of methodologies, we can combine the components and view the effect of moving from one methodology to another. By first analyzing the components, we will know why the methodologies perform as the do. Figures 5.12 shows how each methodology performs with respect to the estimation error. Recall that estimation error decreases when we use logistic regression with a large two-factor interaction coefficient (Figure 5.7), a factorial design with an un-centered mean response (Figure 5.6), or a full factorial design under any condition (Figure 5.5). Figure 5.12 shows the gradual decrease in error as we utilize improved methodologies. As expected, there is a decrease from the current methodology to the intermediate 1 methodology and another small decrease from intermediate 2 to the improved methodology. Finally, the greatest decrease is realized by utilizing the ideal methodology.

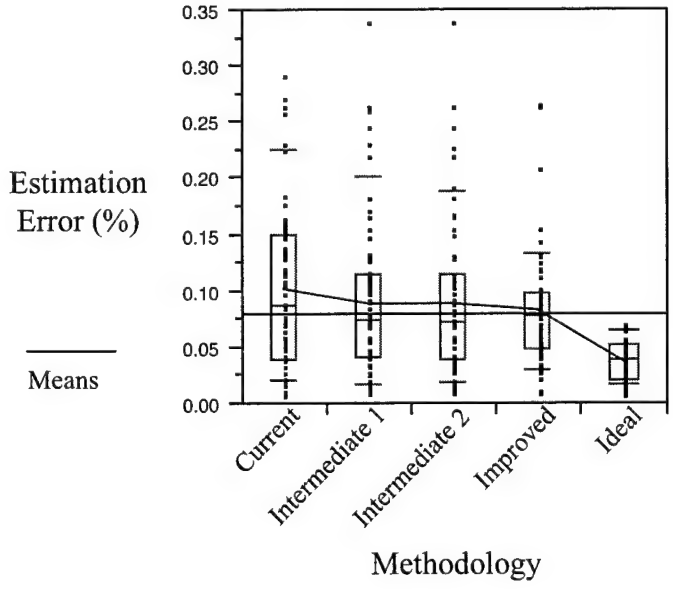

Figure 5.12 Methodology Performance for Estimation Error Response

In Figure 5.13, we see an increase in parameters covered as we improve our methodology. Recall from Figure 5.8 that using a full factorial design or logistic regression both increased our coverage of known parameters. Figure 5.13 shows that there is a significant improvement by adding logistic regression to our methodology, and another improvement by using a full factorial design.

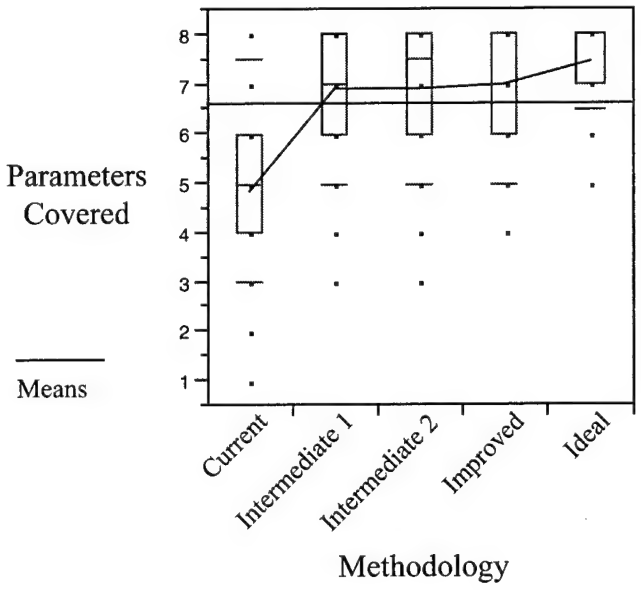

Figure 5.13 Methodology Performance for Parameter Coverage Response

In Figure 5.14, we see the change in interval efficiency due to methodology. Recall in Figures 5.9, 5.10 and 5.11, full factorial experimentation improves interval efficiency, factorial design has the potential to improve efficiency, and logistic regression potentially decreases our efficiency. Figure 5.14 shows that the loss of efficiency due to increasing the width of intervals is not countered unless we utilize a full factorial design.

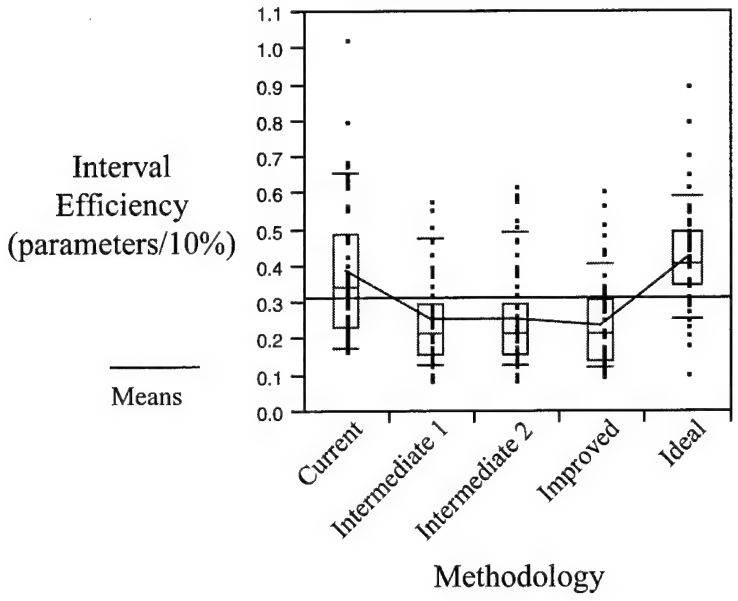

Figure 5.14 Methodology Performance for Interval Efficiency Response

Overall, these graphs demonstrate an improvement in how we address uncertainty in performance evaluations. This research and analysis does not prove that the improved methodology has benefits for all performance evaluations, nor does this research necessarily reflect a realistic quantification of the benefits that can be realized for all evaluations. We only show that a potential for improvement can exist, and where it exists, the benefits are robust to various test outcomes. In ATR performance evaluations, the complexity and breadth of testing makes possible enormous benefits for using the improved methodology. We can, therefore, recommend the experimental design approach for ATR performance evaluations. Our recommendations are presented in chapter 6.

6. CONCLUSIONS AND RECOMMENDATIONS.

In this chapter, we summarize our research objective and assess the results of our analysis. We postulate on the impact of the improved methodology and propose recommendations for the implementation of improvements. Also, we note the scope of our results.

6.1. Review of Research

Recall that our research objective is to show the utility of experimental design in automatic target recognition performance evaluations. We accomplish this by showing a potential for improvement in the current methodology, proposing improvements, and demonstrating the benefits with simulated data.

6.1.1. Current and Improved Methodologies.

The current methodology for ATR evaluations consists of a one-at-a-time test design without provisions for revision, a coarse characterization of image data, and analysis by brute force. We improve this methodology by utilizing a factorial design with the possibility of fractionation, we use iteration and detailed data characterization, and we use logistic regression for analysis and reporting.

6.1.2. Improvements and Benefits.

We use simulated data to demonstrate one possible case where the improved methodology generates better results than the current methodology. We find that the logistic regression technique has the following benefits:

- More efficient identification of significant relationships among variables
- More accurate and more appropriate model of performance using only significant factors
- More accurate confidence interval estimation

We find that increasing the detail of our data characterization has benefits as well:

- More precise prediction of performance
- More accurate estimate of random error
- Potential to reduce estimate of random error

Finally, we list the benefits of using factorial design in our test process:

- Valid prediction of performance for complex conditions
- More efficient use of test resources

Our research shows that the benefits we rationalize in chapter 3 are realized for some data in chapter 4.

6.1.3. Sensitivity of Benefits.

We show via sensitivity analysis that even under entirely different test outcomes, the average effectiveness of the improved methodology is better than the current methodology. Our research does not prove that there are benefits to the improved method for all ATR tests. Rather, we show that the potential for improvements exists for a simple evaluation, even under different test outcomes. It is our opinion that more complex ATR performance data will contain significant interactions among controlled factors as well as many significant background factors. Since our improvements are intended to account for interactions and take advantage of background factors, we expect the benefits will *increase* in more complex tests.

6.2. Conclusions

We conclude that the improved methodology has benefits over the current methodology for tests where there are interactions between variables and background factors that affect performance. We believe the benefits we measure in this research are smaller than the potential benefits for real ATR evaluations. Furthermore, we conclude that there are benefits for tests where the interactions and background factor effects are small or negligible, by merit of this added knowledge and efficiency (i.e.: we *know* there are no interactions and no background effects). We believe these improvements will have a positive effect on ATR performance evaluations. The magnitude of the benefits for various evaluations (and the cost tradeoff) exceeds the scope of our research.

6.2.1. Impact on Performance Evaluations.

The impact of greater accuracy in performance estimation is a higher likelihood of successfully answering test objectives. If our test objective is to evaluate the difference between algorithms, greater estimation accuracy implies a higher likelihood of detecting a difference between the algorithms (if it exists). If the test objective is to characterize algorithm performance across multiple conditions, greater accuracy implies higher likelihood of identifying significant, complex relationships between test factors and performance. In general, we expect an improvement in the ability to distinguish random error from interesting results.

6.2.2. Impact on Test Organization.

The impact of an improved ability to distinguish error from true results is better decisions. Also, increased efficiency in testing can increase the scope of testing or reduce

the cost of testing. We expect the test organization will be able to answer broader test objectives or minimize the cost of answering objectives.

6.2.3. Impact on Automatic Target Recognition Algorithm Acquisition.

We believe there is a potential impact on algorithm acquisition. If we can expand the scope of evaluations we can increase the likelihood of identifying a promising algorithm and thereby shorten the transition time for algorithms and improve the probability of transitioning a good algorithm to operational use. The impact of our improvements cannot guarantee the creation of better algorithms, but we can improve the likelihood that a good algorithm is identified.

6.3. Recommendations

We recommend that the experimental design approach be adopted for ATR performance evaluations. We recommend implementation of all the improvements that are the subject of this research. In this section, we discuss the method for implementing recommendations.

6.3.1. Recommendations for Implementation of Improvements.

The benefits of our improved methodology depend upon proper implementation of recommendations. We recommend that the transition to an experimental design approach be managed by an experimental design practitioner. In our research we do not cover the many assumptions of experimental design so we assert that proper implementation requires an in-depth knowledge of the techniques and methods for their application.

We also recommend a study of possible methods to increase image characterization and reduce the time for data reduction. Taking these steps better facilitates the use of the improved methodology.

6.3.2. Recommendations for Further Research.

In the course of our research, many opportunities for further research have come to our attention. The list below contains the most significant research opportunities.

- A study to research the impact of increased estimation accuracy in true ATR evaluations
- A study to research the cost of utilizing an experimental design approach to testing for ATR evaluations
- A study to research the specific implementation of experimental design with a real ATR evaluation (i.e., a test case)
- A study to research the uncontrolled factors that affect performance in ATR evaluations with emphasis on methods to measure those factors
- A study to research the automation of detailed image characterization

BIBLIOGRAPHY

1. Alsing, Stephen. The Evaluation of Competing Classifiers. Dissertation, AFIT/DSS/ENS/00M-01, Graduate School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 2000.

2. Automatic Target Recognition Working Group. "Applications of Confidence Intervals to ATR Performance Evaluation," ATRWG No. 88-006, October 1988.

3. Automatic Target Recognition Working Group. "Data Collection Guidelines", ATRWG No. 90-002, June 1990.

4. Automatic Target Recognition Working Group. "Requirements for Representative Data in ATR System Development and Evaluation," ATRWG No. 90-001, February 1990.

5. Automatic Target Recognition Working Group. "Target Recognizer Definitions and Performance Measures," ATRWG No. 86-001, February 1986.

6. Box, George E. P. Empirical Model-Building and Response Surfaces, John Wiley & Sons, Inc., 1987.

7. COMPASE home page. "COMPASE center abstract", MS PowerPoint presentation, https://restricted.compase.vdl.afrl.af.mil/, Oct 2000.

8. COMPASE home page. "COMPASE master overview", MS PowerPoint presentation, https://restricted.compase.vdl.afrl.af.mil/, Oct 2000.

9. Department of the Air Force. Developmental Test and Evaluation. Air Force Instruction 99-101, http://afpubs.hq.af.mil, 1 November 1996.

10. Department of the Air Force. Operational Test and Evaluation. Air Force Instruction 99-102, http://afpubs.hq.af.mil, 1 July 1998.

11. Dudgeon, D. E. "ATR Performance Modeling and Estimation", Technical Report 1051, Group 401, MIT Lincoln Laboratory, December 1998.

12. Fleiss, Joseph L. Statistical Methods for Rates and Proportions, 2nd ed. New York, John Wiley & Sons, Inc., 1981.

13. Fries, S., P. Klausmann, U. Jager, G. Saur, D. Willersinn, G. Hofele, and U. Thonnessen. "Evaluation Framework for ATR Algorithms," SPIE Conference on Automatic Target Recognition IX, Vol. 3718, pp. 438 - 448, Orlando, Florida, 1999.

14. Law, Averill M., W. David Kelton. Simulation Modeling and Analysis, 3rd ed. Boston, McGraw Hill, 2000.

15. Li, B., Q. Zheng, S. Der, R. Chellappa, N. Nasrabadi, L. Chan, and L. Wang. "Experimental Evaluation of Neural, Statistical and Model-Based Approaches to FLIR ATR", pp. 388-397, Proc. SPIE, vol 3371-47, Automatic Target Recognition VIII, April 1998.

16. Michel, Jonathan D., Qin Cai, and Keith Drake. "Non-parametric data modeling in SAR image quality assessment," Proceedings of SPIE, Vol. 3370, pp. 166 - 173 Algorithms for Synthetic Aperture Radar Imagery V, 1998.

17. Mossing, J. C. and T. D. Ross. "Evaluation of SAR ATR algorithm performance sensitivity to MSTAR extended operating conditions", pp. 554-556, Proceedings of SPIE, vol 3370, Algorithms for Synthetic Aperture Radar Imagery V, April 1998.

18. Neter, Kutner, Nachtsheim, and Wasserman. Applied Liner Statistical Models, 4th ed., Boston, McGraw-Hill, 1996.

19. Power, Gregory J. "Determining a Confidence Factor for Automatic Target Recognition Based on Image Sequence Quality," Proc. SPIE, Vol. 3370, pp. 156 - 165 Algorithms for Synthetic Aperture Radar Imagery V, 1998.

20. Proceedings of SPIE. "Algorithms for Synthetic Aperture Radar Imagery," volumes I, II, III, IV, V, and VI.

21. Ross, T. D. Lead, Evaluation Methods and Theory, Foundation Areas, Comprehensive Performance Assessment of Sensor Exploitation Center, Air Force Research Laboratory/Sensors Directorate (SNAT), Meeting, 07 Sep 2000.

22. Sims, S. Richard F. "Signal to clutter measurement and ATR performance," Proc. SPIE, Vol. 3371, pp. 398 - 403 Automatic Target Recognition VIII, 1998.

23. Westerkamp, L. A. "ATR Evaluation Best Practices," http://www.mbvlab.wpafb.af.mil/atrwg/committees/evaluation/bprac/bprac.html.

24. Weszka, J., C. Dyer, and A. Rosenfeld. "A Comparative Study of Texture Measures for Terrain Classification," IEEE Transactions on Systems, Man and Cybernetics 6, pp. 269-285, 1976.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
05-03-2001	Master's Thesis	Jun 2000 – Mar 2001

4. TITLE AND SUBTITLE
UTILITY OF EXPERIMENTAL DESIGN IN AUTOMATIC TARGET RECOGNITION PERFORMANCE EVALUATION

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Higdon, James M., Captain, USAF

5d. PROJECT NUMBER
6095

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENS)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/01M-08

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/SN
Attn: Dr. Timothy D. Ross
2241 Avionics Cl Building 620 Rm C3-J75
WPAFB OH 45433
DSN: 785-1115 x4045
Timothy.Ross@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This research investigates current practices in test and evaluation of classification algorithms, and recommends improvements. We scrutinize the evaluation of automatic target recognition algorithms and rationalize the potential for improvements in the accepted methodology. We propose improvements through the use of an experimental design approach to testing. We demonstrate the benefits of improvements by simulating algorithm performance data and using both methodologies to generate evaluation results. The simulated data is varied to test the sensitivity of the benefits to a broad set of outcomes.
The opportunities for improvement are threefold. First, the current practice of "one-at-a-time" factor variation (only one factor is varied in each test condition) fails to capture the effect of multiple factors. Next, the coarse characterization of data misses the opportunity to reduce the estimate of noise in test through the observation of uncontrolled factors. Finally, the lack of advanced data reduction and analysis tools renders analysis and reporting tedious and inefficient. This research addresses these shortcomings and recommends specific remedies through factorial testing, detailed data characterization, and logistic regression. We show how these innovations improve the accuracy and efficiency of automatic target recognition performance evaluation.

15. SUBJECT TERMS
Test and Evaluation, Statistics, Statistical Analysis, Experimental Design, Regression Analysis, Target Recognition, Target Detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Kenneth W. Bauer, ENS kenneth.bauer@afit.edu
U	U	U	UU	116	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 4328

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Form Approved
OMB No. 074-0188